Metadata API Developer Guide

Version 64.0, Summer '25

Last updated: June 27, 2025

© Copyright 2000–2025 Salesforce, Inc. All rights reserved. Salesforce is a registered trademark of Salesforce, Inc., as are other names and marks. Other marks appearing herein may be trademarks of their respective owners.

CONTENTS

INTRODUCTION TO METADATA API 1

Chapter 1: Understanding Metadata API 1

Use Cases for Metadata API 2
- Move Metadata from Production to Your Local File System 3
- Move Metadata Changes to and from a Scratch Org 3
- Move Metadata to a Sandbox at Integration Points 4
- Deploy Metadata to Production 4
- Move Metadata for Production-Level Changes 5
- Make Large Metadata Configuration Changes 5

Release Notes 5
Metadata API Developer Tools 5
Supported Salesforce Editions 6
Metadata API Edit Access 7
Development Platforms 7
Standards Compliance 8
Metadata API Support Policy 8
API End-of-Life Policy 8
Related Resources 9

Chapter 2: Quick Start: Metadata API 10

Prerequisites 11
Step 1: (Optional) Add Metadata Components to an Org Using the UI 11
Step 2: Build a Package.xml Manifest 11
Step 3: Retrieve Components with Metadata API 12

Chapter 3: Build Client Applications 14

Prerequisites 14
Step 1: Generate or Obtain the Web Service WSDLs for Your Organization 15
Step 2: Import the WSDL Files Into Your Development Platform 15
Step 3: Walk Through the Java Sample Code 16

USING METADATA API 25

Chapter 4: Declarative (File-Based) Metadata API Calls 25

Deploying and Retrieving Metadata with the Zip File 25
Slow Deployments 27
Does a Retrieve Job Have a Status of Pending? 27
Sample package.xml Manifest Files 28
Running Tests in a Deployment 34

Running a Subset of Tests in a Deployment . . . 35
Run the Same Tests in Sandbox and Production Deployments . . . 36
Limit on Enqueued Deployments from Apex . . . 36
Maintaining User References . . . 37

Chapter 5: CRUD-Based Metadata API Calls . . . 38

Chapter 6: REST Resources . . . 43

Deploy Metadata with Apex Testing Using REST . . . 45
Check the Status of Your Deployment Using REST Resources . . . 49
Deploy a Recently Validated Component Set Without Tests . . . 51
Cancel a Deployment in Progress Using REST . . . 52
Deploy Metadata with REST API in Salesforce CLI . . . 54

Chapter 7: Error Handling . . . 55

Error Handling for Session Expiration . . . 55

REFERENCE . . . 56

Chapter 8: File-Based Calls . . . 56

deploy() . . . 56
Deleting Components from an Organization . . . 66
checkDeployStatus() . . . 68
cancelDeploy() . . . 69
deployRecentValidation() . . . 70
retrieve() . . . 75
RetrieveRequest . . . 81
checkRetrieveStatus() . . . 82

Chapter 9: CRUD-Based Calls . . . 85

createMetadata() . . . 85
readMetadata() . . . 88
updateMetadata() . . . 89
upsertMetadata() . . . 92
deleteMetadata() . . . 94
renameMetadata() . . . 96
create() . . . 98
delete() . . . 100
update() . . . 102

Chapter 10: Utility Calls . . . 106

checkStatus() . . . 106
describeMetadata() . . . 107
describeValueType() . . . 108
listMetadata() . . . 111

ListMetadataQuery . . . 112

Chapter 11: Result Objects . . . 114

AsyncResult . . . 114
CancelDeployResult . . . 118
DeployResult . . . 118
DescribeMetadataResult . . . 126
DescribeValueTypeResult . . . 127
ReadResult . . . 129
RetrieveResult . . . 129
SaveResult . . . 132
DeleteResult . . . 132
UpsertResult . . . 132
Error . . . 133

Chapter 12: Metadata Types . . . 134

Metadata Components and Types . . . 158
Metadata Coverage Report . . . 160
Unsupported Metadata Types . . . 160
Special Behavior in Metadata API Deployments . . . 160
Data Cloud Metadata Types . . . 160
ActivationPlatform . . . 162
ActivationPlatformActvAttr . . . 167
ActivationPlatformField . . . 167
ActvPfrmDataConnectorS3 . . . 169
ActvPlatformAdncIdentifier . . . 171
ActvPlatformFieldValue . . . 173
AiPluginUtteranceDef . . . 175
CustomerDataPlatformSettings . . . 176
DataConnectorIngestApi . . . 177
DataConnectorS3 . . . 178
DataPackageKitDefinition . . . 179
DataPackageKitObject . . . 182
DataSource . . . 184
DataSourceBundleDefinition . . . 184
DataSourceField . . . 186
DataSourceObject . . . 189
DataSourceTenant . . . 190
DataSrcDataModelFieldMap . . . 191
DataStreamDefinition . . . 192
DataStreamTemplate . . . 194
ExternalDataConnector . . . 198
ExternalDataSource . . . 200
ExternalDataTransportFieldTemplate . . . 207

ExternalDataTranObject . . . 207
ExternalDataTransportObjectTemplate . . . 215
FieldSrcTrgtRelationship . . . 215
InternalDataConnector . . . 217
MarketSegmentDefinition . . . 217
MktCalcInsightObjectDef . . . 219
MktDataTranObject . . . 220
ObjectSourceTargetMap . . . 222
StreamingAppDataConnector . . . 225
AccountRelationshipShareRule . . . 227
AccountingFieldMapping . . . 229
AccountingModelConfig . . . 232
ActionLinkGroupTemplate . . . 236
ActionPlanTemplate . . . 240
ActionableListDefinition . . . 242
AdvAccountForecastSet . . . 248
AffinityScoreDefinition . . . 259
AIApplication . . . 264
AIApplicationConfig . . . 265
AIScoringModelDefinition . . . 266
AIUsecaseDefinition . . . 270
AnalyticsDashboard . . . 281
AnalyticSnapshot . . . 295
AnalyticsVisualization . . . 298
AnalyticsWorkspace . . . 304
AnimationRule . . . 307
AppFrameworkTemplateBundle . . . 309
ArticleType . . . 311
ArticleType Layout . . . 315
ChannelLayout . . . 317
ArticleType CustomField . . . 318
ApexClass . . . 321
ApexComponent . . . 324
ApexEmailNotifications . . . 325
ApexPage . . . 327
ApexTestSuite . . . 329
ApexTrigger . . . 330
AppMenu . . . 332
AppointmentAssignmentPolicy . . . 332
AppointmentSchedulingPolicy . . . 334
ApprovalProcess . . . 337
AssignmentRules . . . 347
AssessmentQuestion . . . 350
AssessmentQuestionSet . . . 355

Audience . . . 357
AuraDefinitionBundle . . . 365
AuthProvider . . . 368
AutoResponseRules . . . 375
BatchCalcJobDefinition . . . 377
BatchProcessJobDefinition . . . 403
BillingSettings . . . 410
BlacklistedConsumer . . . 413
Bot . . . 414
BotBlock . . . 422
BotTemplate . . . 428
BotVersion . . . 435
BrandingSet . . . 466
BriefcaseDefinition . . . 473
BusinessProcessGroup . . . 478
CallCenter . . . 481
CallCenterRoutingMap . . . 486
CallCoachingMediaProvider . . . 488
CampaignInfluenceModel . . . 489
CaseSubjectParticle . . . 491
CareBenefitVerifySettings . . . 492
CareLimitType . . . 495
CareSystemFieldMapping . . . 497
CareProviderSearchConfig . . . 499
CareRequestConfiguration . . . 500
Certificate . . . 503
ChatterExtension . . . 504
ChoiceList . . . 506
ClaimFinancialSettings . . . 509
ClauseCatgConfiguration . . . 511
CleanDataService . . . 513
CMSConnectSource . . . 517
Community (Zone) . . . 525
CommerceSettings . . . 528
CommunityTemplateDefinition . . . 531
CommunityThemeDefinition . . . 537
ConnectedApp . . . 541
ContentAsset . . . 560
ContentTypeBundle . . . 563
ContextDefinition . . . 568
ConversationMessageDefinition . . . 584
ConversationVendorInfo . . . 604
ConvIntelligenceSignalRule . . . 614
CorsWhitelistOrigin . . . 618

CspTrustedSite 620
CustomApplication 623
CustomApplicationComponent 643
CustomFeedFilter 645
CustomFieldDisplay 647
CustomHelpMenuSection 648
CustomIndex 650
CustomLabels 651
Custom Metadata Types (CustomObject) 653
    CustomMetadata 656
CustomNotificationType 662
CustomObject 663
    ActionOverride 678
    BusinessProcess 682
    CompactLayout 683
    CustomField 686
    FieldSet 700
    HistoryRetentionPolicy 701
    Index 702
    ListView 705
    NamedFilter 709
    Picklist (Including Dependent Picklist) 712
    ProfileSearchLayouts 716
    RecordType 717
    SearchLayouts 720
    SharingReason 722
    SharingRecalculation 723
    ValidationRule 724
    WebLink 726
    Metadata Field Types 730
CustomObjectTranslation 733
CustomPageWebLink 743
CustomPermission 746
CustomSite 749
CustomTab 757
CustomValue 761
Dashboard 764
DataCategoryGroup 787
DataObjectSearchIndexConf 792
DataWeaveResource 794
DecisionTable 796
DecisionTableDatasetLink 802
DecisionMatrixDefinition 804
DelegateGroup 810

DigitalExperienceBundle 812
DigitalExperienceBundle: Marketing Workspace Bundle and Folders 818
DigitalExperienceBundle: Site Workspace Bundle and Folders 829
DigitalExperienceConfig 863
DisclosureDefinition 865
DisclosureDefinitionVersion 867
DisclosureType 870
DiscoveryAIModel 873
DiscoveryGoal 878
DiscoveryStory 886
Document 888
DocumentCategory 891
DocumentCategoryDocumentType 892
DocumentChecklistSettings 894
DocumentType 895
DuplicateRule 897
EclairGeoData 902
EmailServicesFunction 904
EmailTemplate 908
EmbeddedServiceBranding 913
EmbeddedServiceConfig 915
EmbeddedServiceFieldService 925
EmbeddedServiceFlowConfig 927
EmbeddedServiceLiveAgent 928
EmbeddedServiceMenuSettings 931
EnablementMeasureDefinition 936
EnablementProgramDefinition 942
EnblProgramTaskSubCategory 954
EntitlementProcess 956
EntitlementTemplate 961
EscalationRules 962
EventDelivery 965
EventRelayConfig 967
EventSubscription 969
ExperienceBundle 971
ExperiencePropertyTypeBundle (Beta) 993
ExplainabilityMsgTemplate 997
ExpressionSetDefinition 1000
ExpressionSetMessageToken 1027
ExpressionSetObjectAlias 1028
ExternalAuthIdentityProvider 1031
ExternalClientApplication 1035
ExternalCredential 1038
ExternalAIModel 1044

ExternalServiceRegistration 1045
ExtlClntAppConfigurablePolicies 1049
ExtlClntAppGlobalOauthSettings 1052
ExtlClntAppMobileConfigurablePolicies (Beta) 1058
ExtlClntAppMobileSettings (Beta) 1059
ExtlClntAppNotificationSettings (Beta) 1060
ExtlClntAppOauthConfigurablePolicies 1061
ExtlClntAppOauthSettings 1068
ExtlClntAppPushConfigurablePolicies (Beta) 1073
ExtlClntAppPushSettings (Beta) 1073
ExtlClntAppSamlConfigurablePolicies 1076
FeatureParameterBoolean 1082
FeatureParameterDate 1084
FeatureParameterInteger 1086
FieldRestrictionRule 1088
FlexiPage 1090
Flow 1111
FlowCategory 1207
FlowDefinition 1208
FlowTest 1209
Folder 1215
    FolderShare 1217
ForecastingFilter 1220
ForecastingFilterCondition 1222
ForecastingSourceDefinition 1224
ForecastingType 1227
ForecastingTypeSource 1230
FuelType 1232
FuelTypeSustnUom 1234
FunctionReference 1237
FundraisingConfig 1238
GatewayProviderPaymentMethodType 1242
GenAiFunction 1244
GenAiPlanner 1252
GenAiPlannerBundle 1260
GenAiPlugin 1271
GenAiPluginInstructionDef 1277
GenAiPromptTemplate 1279
GenAiPromptTemplateActv 1286
GlobalPicklist 1288
GlobalPicklistValue 1289
GlobalValueSet 1292
GlobalValueSetTranslation 1294
Group 1296

HomePageComponent 1297
HomePageLayout 1299
IdentityVerificationProcDef 1300
IdentityVerificationProcDtl 1308
IdentityVerificationProcFld 1314
InboundCertificate 1318
InboundNetworkConnection 1319
IndustriesPricingSettings 1322
IndustriesRatingSettings 1324
InstalledPackage 1325
IntegrationProviderDef 1327
IPAddressRange 1332
KeywordList 1334
Layout 1336
LearningItemType 1357
Letterhead 1360
LightningBolt 1363
LightningComponentBundle 1366
LightningExperienceTheme 1369
LightningMessageChannel 1371
LightningOnboardingConfig 1373
LightningTypeBundle 1374
LiveChatAgentConfig 1379
LiveChatButton 1383
LiveChatDeployment 1388
LiveChatSensitiveDataRule 1390
LoyaltyProgramSetup 1392
ManagedContentType 1407
ManagedEventSubscription (Beta) 1411
ManagedTopics 1414
MarketingAppExtension 1417
MatchingRule 1423
MessagingChannel 1426
Metadata 1437
MetadataWithContent 1438
MfgProgramTemplate 1439
MilestoneType 1441
MlDomain 1443
MLDataDefinition 1447
MLPredictionDefinition 1451
MobileApplicationDetail 1453
MobileSecurityAssignment 1454
MobileSecurityPolicy 1456
MobSecurityCertPinConfig 1460

ModerationRule 1463
MutingPermissionSet 1467
MyDomainDiscoverableLogin 1469
NamedCredential 1471
NavigationMenu 1481
Network 1485
NetworkBranding 1508
NotificationTypeConfig 1511
OauthCustomScope 1513
OauthTokenExchangeHandler 1515
OcrSampleDocument 1520
OcrTemplate 1524
OmniExtTrackingDef 1528
OmniInteractionAccessConfig 1533
OmniInteractionConfig 1535
OmniScript 1537
OmniSupervisorConfig 1547
OmniTrackingGroup 1553
OutboundNetworkConnection 1558
Package 1560
ParticipantRole 1562
PathAssistant 1564
PaymentGatewayProvider 1566
PermissionSet 1568
PermissionSetGroup 1578
PermissionSetLicenseDefinition (Developer Preview) 1580
PersonAccountOwnerPowerUser 1583
PipelineInspMetricConfig 1583
PlatformCachePartition 1585
PlatformEventChannel 1588
PlatformEventChannelMember 1591
PlatformEventSubscriberConfig 1595
Portal 1598
PortalDelegablePermissionSet 1601
PostTemplate 1603
ProductAttributeSet 1604
PresenceDeclineReason 1604
PresenceUserConfig 1605
PricingActionParameters 1608
PricingRecipe 1611
Profile 1616
ProfileActionOverride 1633
ProfilePasswordPolicy 1635
ProfileSessionSetting 1637

Prompt 1639
PublicKeyCertificate 1650
PublicKeyCertificateSet 1652
Queue 1655
QueueRoutingConfig 1659
QuickAction 1662
RedirectWhitelistUrl 1668
RecommendationStrategy 1669
RecordActionDeployment 1700
RecordAggregationDefinition 1705
RecordAlertCategory 1712
RegisteredExternalService 1714
ReferencedDashboard 1717
RelatedRecordAssocCriteria 1719
RelationshipGraphDefinition 1722
RemoteSiteSetting 1726
Report 1727
ReportType 1757
RestrictionRule 1761
Role 1764
RoleOrTerritory 1765
SalesWorkQueueSettings 1767
SamlSsoConfig 1768
SchedulingObjective 1772
SchedulingRule 1774
Scontrol 1777
SearchCustomization 1780
SearchOrgWideObjectConfig 1785
ServiceAISetupDefinition 1787
ServiceAISetupField 1789
ServiceChannel 1790
ServicePresenceStatus 1793
ServiceProcess 1794
Settings 1801
- AccountSettings 1811
- AccountInsightsSettings 1813
- AccountIntelligenceSettings 1814
- AccountingSettings 1816
- ActionsSettings 1818
- ActivitiesSettings 1819
- AddressSettings 1823
- AIReplyRecommendationsSettings 1827
- AgentforceForDevelopersSettings 1828
- AnalyticsSettings 1829

ApexSettings . . . 1840
AppAnalyticsSettings . . . 1842
AppExperienceSettings . . . 1844
AssociationEngineSettings . . . 1845
AutomatedContactsSettings . . . 1846
BotSettings . . . 1848
BranchManagementSettings . . . 1848
BusinessHoursSettings . . . 1850
CampaignSettings . . . 1854
CaseSettings . . . 1856
ChatterAnswersSettings . . . 1867
ChatterEmailsMDSettings . . . 1869
ChatterSettings . . . 1870
CodeBuilderSettings . . . 1874
CollectionsDashboardSettings . . . 1875
CommunitiesSettings . . . 1876
CompanySettings . . . 1879
ConnectedAppSettings . . . 1880
ContentSettings . . . 1881
ContractSettings . . . 1884
ConversationalIntelligenceSettings . . . 1885
ConversationChannelDefinition . . . 1887
CurrencySettings . . . 1893
CustomAddressFieldSettings . . . 1895
DataDotComSettings . . . 1897
DataImportManagementSettings . . . 1898
DeploymentSettings . . . 1899
DevHubSettings . . . 1900
DocumentGenerationSetting . . . 1902
DynamicFormsSettings . . . 1903
EACSettings . . . 1905
EinsteinAgentSettings . . . 1908
EinsteinGptSettings . . . 1909
EmailAdministrationSettings . . . 1910
EmailIntegrationSettings . . . 1914
EmailTemplateSettings . . . 1917
EmployeeUserSettings . . . 1918
EnhancedNotesSettings . . . 1920
EncryptionKeySettings . . . 1921
EntitlementSettings . . . 1923
EventSettings . . . 1925
ExperienceBundleSettings . . . 1927
ExternalClientAppSettings . . . 1928
ExternalServicesSettings . . . 1930

FieldServiceSettings 1931
FilesConnectSettings 1938
FileUploadAndDownloadSecuritySettings 1939
FlowSettings 1943
ForecastingObjectListSettings 1949
ForecastingSettings 1954
HighVelocitySalesSettings 1967
IdeasSettings 1970
IdentityProviderSettings 1971
IframeWhiteListUrlSettings 1973
IncidentMgmtSettings 1974
IndustriesEinsteinFeatureSettings 1977
IndustriesLoyaltySettings 1978
IndustriesSettings 1981
InterestTaggingSettings 1995
InventorySettings 1997
InvLatePymntRiskCalcSettings 1998
InvocableActionSettings 1999
KnowledgeSettings 2000
LanguageSettings 2007
LeadConfigSettings 2009
LeadConvertSettings 2011
LiveAgentSettings 2014
LightningExperienceSettings 2015
LiveMessageSettings 2019
MacroSettings 2020
MailMergeSettings 2021
MapAndLocationSettings 2022
MeetingsSettings 2023
MobileSettings 2025
MyDomainSettings 2028
MfgServiceConsoleSettings 2036
NameSettings 2037
NotificationsSettings 2038
OauthOidcSettings 2039
ObjectHierarchyRelationship 2041
ObjectLinkingSettings (Beta) 2046
OmniChannelSettings 2047
OpportunityInsightsSettings 2048
OpportunitySettings 2050
OpportunityScoreSettings 2054
OrderManagementSettings 2055
OrderSettings 2056
OrgPreferenceSettings 2058

OrgSettings 2060
PartyDataModelSettings 2062
PardotSettings 2063
PardotEinsteinSettings 2065
PathAssistantSettings 2066
PaymentsSettings 2068
PicklistSettings 2069
PlatformEncryptionSettings 2070
PlatformEventSettings 2072
PredictionBuilderSettings 2073
PrivacySettings 2074
ProcessFlowMigration 2075
ProductSettings 2077
QuoteSettings 2078
RealTimeEventSettings 2079
RecordPageSettings 2081
RetailExecutionSettings 2082
SalesAgreementSettings 2083
SandboxSettings 2088
SchemaSettings 2089
SearchSettings 2090
SecuritySettings 2094
ServiceCloudVoiceSettings 2112
ServiceSetupAssistantSettings 2113
SharingSettings 2113
SiteSettings 2117
SocialCustomerServiceSettings 2119
SocialProfileSettings 2121
SourceTrackingSettings (Beta) 2122
SubscriptionManagementSettings 2123
SurveySettings 2125
Territory2Settings 2126
TrailheadSettings 2130
TrialOrgSettings 2131
UserEngagementSettings 2132
UserInterfaceSettings 2136
UserManagementSettings 2139
VoiceSettings 2143
WarrantyLifeCycleMgmtSettings 2145
WorkDotComSettings 2147
WorkforceEngagementSettings 2148
SharedTo 2150
SharingBaseRule 2153
SharingRules 2154

BaseSharingRule . . . 2162
CriteriaBasedSharingRule . . . 2163
OwnerSharingRule . . . 2169
SharingSet . . . 2175
SiteDotCom . . . 2179
Skill . . . 2180
StandardValueSet . . . 2182
StandardValueSetTranslation . . . 2184
StaticResource . . . 2185
SustainabilityUom . . . 2186
SustnUomConversion . . . 2189
SvcCatalogCategory . . . 2192
SvcCatalogFulfillmentFlow . . . 2194
SvcCatalogItemDef . . . 2198
SynonymDictionary . . . 2207
Territory . . . 2209
Territory2 . . . 2210
Territory2Model . . . 2215
Territory2Rule . . . 2217
Territory2Type . . . 2220
TimelineObjectDefinition . . . 2221
TimeSheetTemplate . . . 2223
TopicsForObjects . . . 2226
TransactionSecurityPolicy . . . 2227
Translations . . . 2233
UiFormatSpecificationSet . . . 2253
UIObjectRelationConfig . . . 2258
UserAccessPolicy . . . 2263
UserAuthCertificate . . . 2269
UserCriteria . . . 2270
UserProfileSearchScope . . . 2271
UserProvisioningConfig . . . 2271
VirtualVisitConfig . . . 2273
WaveAnalyticAssetCollection . . . 2278
WaveApplication . . . 2280
WaveComponent . . . 2281
WaveDataflow . . . 2282
WaveDashboard . . . 2283
WaveDataset . . . 2284
WaveLens . . . 2285
WaveRecipe . . . 2287
WaveTemplateBundle . . . 2288
WaveXmd . . . 2289
WebStoreBundle . . . 2298

WebStoreTemplate 2298
Workflow 2302
WorkSkillRouting 2317

Chapter 13: Headers 2320

AllOrNoneHeader 2320
CallOptions 2322
DebuggingHeader 2323
SessionHeader 2325

APPENDICES 2326

Appendix A: CustomObjectTranslation Language Support: Fully Supported Languages 2326

Appendix B: CustomObjectTranslation Language Support: End-User Languages 2331

Appendix C: StandardValueSet Names and Standard Picklist Fields . . 2338

INDEX 2360

INTRODUCTION TO METADATA API

CHAPTER 1 Understanding Metadata API

Salesforce Metadata

Metadata is data that describes other data. To understand how Salesforce defines metadata, contrast business data with Salesforce metadata. Business data includes the records that directly correspond to your company's business such as an address, account, or product. Salesforce metadata describes the schema, process, presentation, authorization, and general configuration of your Salesforce org.

To contrast Salesforce metadata with business data, first examine how schema metadata describes the properties of business data. For example, the Salesforce standard object Address has schema metadata and business data. Address fields such as `Address Type`, `City`, and `Postal Code`, are all schema metadata. The corresponding values in each field, such as Mailing address, Chicago, IL, and 60106, are all data. While personally identifiable information (PII) is usually found in business data, metadata can also include PII, such as custom object names, report names, etc.

Metadata in Salesforce also defines how your org functions. For example, process metadata describes what happens when a user presses the Save button. Presentation metadata concerns the layout of your org, and authorization metadata determines user access. Salesforce metadata also describes your org's general configuration. For example, you can configure Chatter to block emoticons in posts.

Metadata API works with metadata types and components. A metadata type defines the structure of application metadata. A metadata component is an instance of a metadata type. The fields and values of a metadata type are all metadata. For example, the metadata type CustomTab on page 757 represents a custom tab that displays content. The CustomTab field `hasSidebar` indicates if the tab is on the sidebar panel, which is an example of metadata determining presentation. Metadata types like CustomTab build the metadata model that describe how your org is structured, displayed, or functions. Use Metadata API to develop customizations and build tools that manage the metadata model, not the data itself.

Metadata API Functionality

The main purpose of Metadata API is to move metadata between Salesforce orgs during the development process. Use Metadata API to deploy, retrieve, create, update, or delete customization information, such as custom object definitions and page layouts. Metadata API doesn't work directly with business data. To create, retrieve, update, or delete records such as accounts or leads, use SOAP API or REST API.

You can move metadata with one of two ways. The first method is with Metadata API `deploy()` and `retrieve()` calls. Admins often use the `deploy()` and `retrieve()` calls to move the full metadata model. These calls are best fit for the final stages of development, such as deploying tested customizations to the production org.

The second method is source push and pull commands that move only changes in metadata. These commands use source tracking, which makes them friendlier for developers and better for intermediary stages of development.

SEE ALSO:

- Metadata Components and Types
- Deploying and Retrieving Metadata
- source Commands

Use Cases for Metadata API

Use Metadata API to move metadata between orgs during the development cycle. Metadata API is also used for deploying large metadata configuration changes from development.

To understand how to use Metadata API, let's imagine you're a Salesforce developer at Zephyrus Relocation Services. Zephyrus is a talent-mobility firm that helps companies develop processes for domestic and international employee relocation. Zephyrus is expanding into Asia and South America and wants to add orientation services for both regions. Orientation services include in-country assistance in housing and school searches, area tours, and transportation information.

Your development team must add these new orientation services to their existing org. Products such as in-country orientations are objects that can be customized in Salesforce. When you add objects and customize your org, you change its metadata. The development process of creating a custom product is where Metadata API can help.

Use Metadata API in the Development Process

Currently, Zephyrus has production metadata and orientation services tailored to other countries. To begin building the new product customizations, you need the existing configurations from Zephyrus' production Salesforce org in a separate repository. The configuration of the production org is all metadata. To save production metadata in a repository, move the metadata from the Zephyrus production org to your local file system.

Move Metadata from Production to Your Local File System on page 3

To make development changes without affecting your existing configurations, use Metadata API to move metadata to your local file system. Next, push metadata from your local file system to a shareable repository for development.

With all the Zephyrus metadata retrieved, you can develop locally or in a scratch org. Scratch orgs are disposable Salesforce environments with no data. Many developers use both tools together. Loading files and making changes are much faster locally than doing so in a scratch org. Developers often build customizations on their local file system and run tests in a scratch org. Move changes between your local file system and scratch org as you test and develop.

Move Metadata Changes to and from a Scratch Org on page 3

You can use a scratch org along with your local file system to develop and test changes to metadata. To move the changes you make locally to and from the scratch org, use Metadata API.

The rest of the Zephyrus development team has their own customizations. After developing and testing on your own, it's time for the team to integrate changes and run tests in sandboxes. Sandboxes are development environments used for developing and testing integrations.

Move Metadata to a Sandbox at Integration Points on page 4

During development, use Metadata API to move metadata to sandboxes for integrating changes, testing, and collaborating with your team.

After your team builds the orientation service customization and completes testing, deploy these components to production with Metadata API.

Deploy Metadata to Production on page 4

In the final step of the development cycle, move customizations from a source control system such as Git into production with Metadata API.

Other Use Cases

You can use Metadata API for larger changes in Salesforce, such as splitting and merging production orgs.

For example, Zephyrus wants to split the company into two divisions, one that specializes in domestic relocation and another for international relocation. In this case, you split Zephyrus' Salesforce org and decide which metadata belongs in which org. Metadata API can move metadata to the new org.

Then, let's say Zephyrus acquires Apollo Global Relocation and both companies use Salesforce. To consolidate information, you use Metadata API to merge the Apollo org into the Zephyrus org.

Move Metadata for Production-Level Changes on page 5

Use Metadata API to move metadata during large changes, such as merging or splitting Salesforce orgs.

You can use Metadata API to make configuration changes during the development process that are too large for alternative API calls. For example, Zephyrus supports many languages for their global clients. To translate different languages for your objects, you include an object translation file for each language.

Make Large Metadata Configuration Changes on page 5

Metadata is better suited than other APIs for deploying large changes to your org.

Move Metadata from Production to Your Local File System

To make development changes without affecting your existing configurations, use Metadata API to move metadata to your local file system. Next, push metadata from your local file system to a shareable repository for development.

When you build customizations on Salesforce, you must preserve the functionality of your existing org during the development cycle. To build customizations without affecting your production org, save your production metadata in a version control system. Git integrates best with SFDX tools.

First, move the required metadata from the production org to your local file system. To move metadata to your local machine, use a retrieve call instead of a source pull. Next, push your files to a repository that is accessible to your team members with Git commands. The repository is now the original source of production metadata for your team's development cycle.

Now that your production metadata is stored in a repository, move the necessary metadata back to your local file system to begin development work.

SEE ALSO:

- retrieve()
- source Commands

Move Metadata Changes to and from a Scratch Org

Use a scratch org to develop and test changes to metadata. You can perform your development within or outside the scratch org using Salesforce CLI or Salesforce Extensions for VS Code, which leverage the power of Metadata API.

Scratch orgs are created empty so that developers can specify the exact metadata and data to include from the source control system. The lifespan of a scratch org is indicated during creation, 1–30 days. They're ephemeral to ensure the source of truth is always the source control system, and not the org itself.

You can move metadata from the source control system or to the scratch org using Salesforce CLI. Because scratch orgs use source tracking to identify changes, the CLI is the most efficient way to move metadata between your local repository and the scratch org. Continue to iterate through this process of moving metadata between your local file system and your scratch org until development is complete.

SEE ALSO:

- Push Source to the Scratch Org
- Pull Source from the Scratch Org to Your Project

Move Metadata to a Sandbox at Integration Points

During development, use Metadata API to move metadata to sandboxes for integrating changes, testing, and collaborating with your team.

After developing on your own in a scratch org or your local file system, combine work from your team at integration points in a sandbox. Sandboxes are development environments that you can use to integrate and test changes from multiple developers. Admins often create and assign sandboxes. To create a sandbox on the Salesforce UI, navigate to Setup. Next, in the Quick Find box, search for sandboxes.

You have several levels of sandboxes to choose from with differing amounts of data. The Developer sandbox and Developer Pro sandbox are development environments where you build customizations and test changes on fictional data. The Partial Copy sandbox and Full sandbox are testing environments loaded with copies of production data. Move metadata to different sandboxes with a Metadata API deploy command depending on your development and testing needs.

Outside of Metadata API, admins typically use change sets to send customizations from one sandbox to another. Unlike Metadata API calls, you must build change sets manually. To add components to a continuous integration system more easily, you can automate Metadata API calls on Salesforce CLI.

SEE ALSO:

- Sandbox Types and Templates
- Change Sets
- Continuous Integration

Deploy Metadata to Production

In the final step of the development cycle, move customizations from a source control system such as Git into production with Metadata API.

When your team finishes integration tests and is ready to deploy to production, move the completed customizations from a local environment to the repository. For the release, move metadata from the repository to production by pulling the updated repository back to the local environment with Git commands. Next, deploy metadata to production with Metadata API deploy call.

Moving metadata to production requires a deploy call instead of a push command because the deploy call deploys the entire metadata model and not just changes in the metadata.

Deploy Recent Validation

A regular deploy call executes automated Apex tests that can take a long time to complete. To skip tests for validated components and deploy components to production quickly, use the deploy recent validation option.

SEE ALSO:

- deploy()
- force:source:push Command
- Deploy a Recently Validated Component Set Without Tests

Move Metadata for Production-Level Changes

Use Metadata API to move metadata during large changes, such as merging or splitting Salesforce orgs.

To split an org, first retrieve the metadata to be moved. Then, use a deploy call to push those configurations to the new org. Similarly, to merge two orgs, retrieve existing metadata from one org. Next, use a deploy call to migrate metadata from one org to another.

SEE ALSO:

- retrieve()
- deploy()

Make Large Metadata Configuration Changes

Metadata API is better suited than other APIs for deploying large changes to your Salesforce org.

Metadata API `deploy()` and `retrieve()` calls are file-based and therefore asynchronous. With synchronous commands, large configuration changes require unreasonably long load times. Instead, deploy and retrieve calls begin an asynchronous process that notifies you when it's complete. Because file-based calls are asynchronous, Metadata API can also handle a queue of deploy requests.

SEE ALSO:

- deploy()
- retrieve()

Metadata API Release Notes

Use the Salesforce Release Notes to learn about the most recent updates and changes to Metadata API.

For updates and changes that impact the Salesforce Platform, including Metadata API, see the API Release Notes.

For new, changed, and deprecated metadata types and other changes specific to Metadata API, see Metadata API in the Salesforce Release Notes.

Metadata API Developer Tools

Use the Salesforce Extensions for Visual Studio Code on Salesforce CLI to access Metadata API commands. Salesforce CLI and the Salesforce Extensions for Visual Studio Code streamline the process of using Metadata API.

The easiest way to access the functionality in Metadata API is to use the Salesforce Extensions for Visual Studio Code or Salesforce CLI. Both tools are built on top of Metadata API and use the standard tools to simplify working with Metadata API.

- The Salesforce Extensions for Visual Studio Code includes tools for developing on the Salesforce platform in the lightweight, extensible VS Code editor. These tools provide features for working with development orgs (scratch orgs, sandboxes, and DE orgs), Apex, Aura components, and Visualforce.
- Salesforce CLI is ideal if you use scripting or the command line for moving metadata between a local directory and a Salesforce org.

For more information about the Salesforce Extensions for Visual Studio Code or Salesforce CLI, see Salesforce Tools and Toolkits.

If you prefer to build your own client applications, the underlying calls of Metadata API have been exposed for you to use directly. This guide gives you more information about working directly with Metadata API.

You can use the Metadata API to manage setup and customization information (metadata) in Salesforce. For example:

- Export customizations as XML metadata files. See Working with the Zip File and `retrieve()`.
- Migrate configuration changes between orgs. See `deploy()` and `retrieve()`.
- Modify existing customizations using XML metadata files. See `deploy()` and `retrieve()`.
- Manage customizations programmatically. See CRUD-Based Metadata Development.

You can modify metadata in test orgs in Developer Edition or sandboxes, and then deploy tested changes to production orgs in Enterprise, Unlimited, or Performance Editions. You can also create scripts to populate a new org with your custom objects, custom fields, and other components.

SEE ALSO:

Deploying and Retrieving Metadata

Supported Salesforce Editions

To use Metadata API, your organization must use **Enterprise Edition**, **Unlimited Edition**, **Performance Edition**, or **Developer Edition**. If you're an existing Salesforce customer and want to upgrade to Enterprise, Unlimited, or Performance Edition, contact your account representative.

We strongly recommend that you use a sandbox, which is an exact replica of your production organization. Enterprise, Unlimited, and Performance Editions come with free developer sandboxes. For more information, see `http://www.salesforce.com/platform/cloud-infrastructure/sandbox.jsp`.

Alternatively, you can use a Developer Edition (DE) org. A DE org provides access to all features that are available with Enterprise Edition, but is limited by the number of users and the amount of storage space. A DE org isn't a copy of your production org/ It provides an environment where you can build and test your solutions without affecting your organization's data. Developer Edition accounts are available for free at `https://developer.salesforce.com/signup`.

Note: A metadata component must be visible in the org for Metadata API to act on it. Also, a user must have the API Enabled permission to have access to metadata components.

Metadata API Access for Professional Edition

ISV partners can request Metadata API access to Professional Edition orgs for apps that have passed AppExchange Security Review. Access is granted through an API token (client ID). This special key enables the app to make Metadata API calls to customers' Professional Edition orgs.

As an ISV partner, you can request Metadata API access by following these steps.

1. Submit your app for security review. See Steps in the Security Review in the *ISVForce Guide*.
2. After your app is approved, log a case in the Partner Community in **AppExchange and Feature Requests** > **API Token Request**, and specify SOAP for the type of token.

To make calls to the Metadata API, append the API token to the CallOptions SOAP header in your calls.

Metadata API Edit Access

To use Metadata API, a user must have these things.

- One of these editions: Enterprise, Unlimited, or Developer
- Either the Modify Metadata Through Metadata API Functions OR Modify All Data permission
- Permission that enables use of the feature supported by the metadata that they want to modify
- Permission that enables their deployment tool, such as Salesforce CLI, change sets, or the Ant Migration Tool

With the Modify Metadata Through Metadata API Functions permission, a user can access and edit metadata via Metadata API as long as the user has any additional permission needed to access certain metadata types. This additional permission information is listed in the *Metadata API Developer Guide* for each metadata type. With the Modify All Data permission, a user can access and edit all data.

The Modify Metadata Through Metadata API Functions permission doesn't affect direct customization of metadata using Setup UI pages because those pages don't use Metadata API for updates.

Some metadata, such as Apex, executes in system context, so be careful how you delegate the Modify Metadata Through Metadata API Functions permission. The Modify Metadata Through Metadata API Functions permission allows deployment of Apex metadata, but it doesn't allow certain Apex development and debugging features that still require the Modify All Data permission.

The Modify Metadata Through Metadata API Functions permission is enabled automatically when either the Deploy Change Sets OR Author Apex permission is selected.

When the Manage Prompts user permission and the Modify Metadata Through Metadata API Functions permission are combined, users can manage In-App Guidance in Lightning Experience.

Development Platforms

Metadata API supports both file-based and CRUD-based development.

File-Based Development

The declarative or file-based asynchronous Metadata API `deploy()` and `retrieve()` operations deploy or retrieve a `.zip` file that holds components in a set of folders, and a manifest file named `package.xml`. For more information, see Deploying and Retrieving Metadata on page 25. The easiest way to access the file-based functionality is to use the Salesforce Extensions for Visual Studio Code or the Ant Migration Tool.

CRUD-Based Development

The CRUD Metadata API calls act upon the metadata components in a manner similar to the way synchronous API calls in the *enterprise WSDL* act upon objects. For more information about the enterprise WSDL, see the *SOAP API Developer Guide*.

Standards Compliance

Metadata API is implemented to comply with the following specifications:

Standard Name	Website
Simple Object Access Protocol (SOAP) 1.1	http://www.w3.org/TR/2000/NOTE-SOAP-20000508/
Web Service Description Language (WSDL) 1.1	http://www.w3.org/TR/2001/NOTE-wsdl-20010315
WS-I Basic Profile 1.1	http://www.ws-i.org/Profiles/BasicProfile-1.1-2004-08-24.html

Metadata API Support Policy

Salesforce supports previous versions of Metadata API. However, your new client applications should use the most recent version of the Lightning Platform Metadata API WSDL file to fully exploit the benefits of richer features and greater efficiency.

Backward Compatibility

Salesforce strives to make backward compatibility easy when using the Lightning Platform.

Each new Salesforce release consists of two components:

- A new release of platform software that resides on Salesforce systems
- A new version of the API

For example, the Spring '07 release included API version 9.0 and the Summer '07 release included API version 10.0.

We maintain support for each API version across releases of the platform software. The API is backward compatible in that an application created to work with a given API version will continue to work with that same API version in future platform software releases.

Salesforce does not guarantee that an application written against one API version will work with future API versions: Changes in method signatures and data representations are often required as we continue to enhance the API. However, we strive to keep the API consistent from version to version with minimal, if any, changes required to port applications to newer API versions.

For example, an application written using API version 9.0, which shipped with the Spring '07 release, will continue to work with API version 9.0 on the Summer '07 release, and on future releases beyond that. However, that same application might not work with API version 10.0 without modifications to the application.

API End-of-Life Policy

See which Metadata REST and SOAP API versions are supported, unsupported, or unavailable.

Salesforce is committed to supporting each API version for a minimum of 3 years from the date of first release. To improve the quality and performance of the API, versions that are over 3 years old sometimes are no longer supported.

Salesforce notifies customers who use an API version scheduled for deprecation at least 1 year before support for the version ends.

Salesforce API Versions	Version Support Status	Version Retirement Info
Versions 31.0 through 64.0	Supported.	

Salesforce API Versions	Version Support Status	Version Retirement Info
Versions 21.0 through 30.0	As of Summer '22, these versions have been deprecated and no longer supported by Salesforce. Starting Summer '25, these versions will be retired and unavailable.	Salesforce Platform API Versions 21.0 through 30.0 Retirement
Versions 7.0 through 20.0	As of Summer '22, these versions are retired and unavailable.	Salesforce Platform API Versions 7.0 through 20.0 Retirement

If you request any resource or use an operation from a retired API version, REST API returns the `410:GONE` error code.

If you request any resource or use an operation from a retired API version, SOAP API returns `500:UNSUPPORTED_API_VERSION` error code.

To identify requests made from old or unsupported API versions, use the API Total Usage event type.

Related Resources

The Salesforce developer website provides a full suite of developer toolkits, sample code, sample SOAP messages, community-based support, and other resources to help you with your development projects. Be sure to visit `https://developer.salesforce.com/page/Getting_Started` for more information, or visit `https://developer.salesforce.com/signup` to sign up for a free Developer Edition account.

You can visit these websites to find out more about Salesforce applications:

- Salesforce Developers provides useful information for developers.
- Salesforce for information about the Salesforce application.
- Lightning Platform AppExchange for access to apps created for Salesforce.
- Trailblazer Community for services to ensure Salesforce customer success.

CHAPTER 2 Quick Start: Metadata API

In this chapter ...

- Prerequisites
- Step 1: (Optional) Add Metadata Components to an Org Using the UI
- Step 2: Build a Package.xml Manifest
- Step 3: Retrieve Components with Metadata API

Get started with Metadata API by retrieving a small set of metadata components from your org on the Salesforce CLI.

Resources for Beginner Developers

If you're a beginner developer and haven't used Salesforce CLI before, learn how to set up your environment and practice with a sample application. These Trailheads guide you through setup with SFDX and introduce you to Metadata API.

App Development with Salesforce DX

Walk through setting up your environment and developing with Salesforce CLI using the Dreamhouse sample app. After you add a feature to your Dreamhouse app, you deploy metadata to your Dev Hub org with Salesforce CLI.

Package.xml Metadata Management

Learn more about metadata and package.xml files. Build a package.xml file to deploy changes from a scratch org to your Trailhead Playground.

Quick Start for Developing with Metadata API

If you have some experience in Salesforce development but want to get started with Metadata API, use this quick start. The quick start walks you through a retrieval of metadata components, which is the first step of the development process.

SEE ALSO:

Move Metadata from Production to Your Local File System

Prerequisites

Complete these prerequisites before you start developing with Metadata API.

- To access Metadata API through the command line, install Salesforce CLI.
- To create a development environment, sign up for Salesforce Developer Edition. A Developer Edition org is a free development environment for building and testing solutions independent of production data.
- Install Salesforce Extensions for Visual Studio Code. These tools provide features for working with development orgs (scratch orgs, sandboxes, and DE orgs), Apex, Aura components, and Visualforce.
- Confirm that you have API Enabled permission and Modify Metadata Through Metadata API Functions permission or Modify All Data permission. If you don't have these permissions set, modify your metadata permissions.
- Enable Dev Hub in your org. Dev Hub allows you to create and manage scratch orgs so that you can develop without affecting your production data or metadata.
- To allow access to protected resources such as production data and metadata, authorize your org .

Step 1: (Optional) Add Metadata Components to an Org Using the UI

If you're starting with a new practice org that doesn't have customizations, you only have standard metadata that can't be retrieved. To use the Metadata API retrieve call, add a component on the Salesforce UI to your practice org. If you're working on an existing project, you already have components to retrieve and can skip this step.

1. From Setup, click **Create**.
2. Select **Custom Object**.
3. Enter an arbitrary name for **Label** and **Plural Label**.
4. Save the component.

Step 2: Build a Package.xml Manifest

The package.xml manifest file lists the components to retrieve from your org.

Package.xml Manifest Structure

The package.xml manifest uses Extensible Markup Language (XML) to identify and migrate metadata components. The basic framework of the package.xml manifest is built with `<types>` elements. A `<types>` element specifies which metadata type you want to work with. You can add multiple `<types>` to a package.xml file.

Inside the `<types>` element is the `<name>` element and the `<members>` element. The `<members>` element selects for individual components of a specific type, and the <name> element selects for metadata component types. To work with a specific component, input the *`fullName`* of the component in the `<members>` element.

For example, to retrieve Account components, add Account in the `<members>` element and CustomObject in the `<name>` element in your package.xml. When you issue a retrieve call, you retrieve only the Account component from your org.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>Account</members>
```

```
        <name>CustomObject</name>
    </types>
    <version>64.0</version>
</Package>
```

Retrieve Custom Objects

To retrieve all components of a metadata type, you don't specify the *fullName* of a component. Instead, use the wildcard character * (asterisk) in the `<members>` tag. Some components, such as standard objects, don't support * (asterisk) as a specifier.

To retrieve all custom objects from your org:

1. (Optional) If you do not have a project folder, use Salesforce CLI to create a new directory that organizes your project. Run this command with your specified project name:

   ```
   sf project generate --name YourProjectName
   ```

2. Create a file called package.xml in your project.
3. In your text editor, open the file and paste in this script:

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>CustomObject</name>
    </types>
    <version>64.0</version>
</Package>
```

Now you have a package.xml file that we can use to retrieve all custom objects. When you develop more on your own, you can retrieve more components from your org with multiple `<types>` elements.

SEE ALSO:

Sample package.xml Manifest Files

Deploying and Retrieving Metadata with the Zip File

Step 3: Retrieve Components with Metadata API

With Salesforce CLI, retrieve a file representation of the specified components in your package.xml manifest.

Two Options for Metadata API Retrieve

You can use one of two commands to retrieve metadata components.

1. To retrieve the components specified in your package.xml manifest, issue a retrieve call using a Salesforce CLI command. On the command line, run this call with the appropriate file path:

   ```
   sf project retrieve start --manifest path/to/package.xml
   ```

   Metadata `retrieve()` is an asynchronous, file-based command. You can issue multiple retrieve or deploy requests that run on their own when resources are available.

With this command, you send a request to retrieve all custom objects as specified in your package.xml manifest. Your requests are queued until our systems are ready to process your retrieve call. After your request is dequeued, your retrieve call is run. The client checks the status of the retrieve and notifies you when the call is complete. The call returns a file representation of the chosen components. When you use Salesforce CLI to issue a retrieve call, all these processes are automated.

The `project retrieve start` command allows for source tracking. Source tracking includes information about which revision you're working on and when the last changes were made, which makes source commands more developer-friendly. To use source tracking, ensure that it's enabled in your org.

2. Alternatively, run this command in your terminal:

```
sf project retrieve start --manifest path/to/package.xml --target-metadata-dir
path/to/retrieve/dir
```

This command retrieves components in mdapi format rather than source format, and doesn't allow for source tracking. In practice, admins use mdapi commands more often because the commands don't include source tracking.

SEE ALSO:

retrieve()

source Commands

Source Tracking

mdapi Commands

CHAPTER 3 Build Client Applications for Metadata API

Use Metadata API to retrieve, deploy, create, update, or delete customizations for your org. The most common use is to migrate changes from a sandbox or testing org to your production environment. Metadata API is intended for managing customizations and for building tools that can manage the metadata model, not the data itself.

Salesforce CLI automates the underlying calls of Metadata API. However, you can use these calls directly with your own client application. This guide gives you all the information you require to start writing applications that directly use Metadata API to manage customizations for your organization. It shows you how to get started with File-Based Development. For an example of CRUD-Based Development, see Java Sample for CRUD-Based Development with Synchronous Calls.

Prerequisites

Make sure that you complete these prerequisites before you start using Metadata API.

- Create a development environment.

  We strongly recommend that you use a sandbox, which is an exact replica of your production organization. Enterprise, Unlimited, and Performance Editions come with free developer sandboxes. For more information, see `http://www.salesforce.com/platform/cloud-infrastructure/sandbox.jsp`.

  Alternatively, you can use a Developer Edition (DE) org. A DE org provides access to all features that are available with Enterprise Edition, but is limited by the number of users and the amount of storage space. A DE org isn't a copy of your production org/ It provides an environment where you can build and test your solutions without affecting your organization's data. Developer Edition accounts are available for free at `https://developer.salesforce.com/signup`.

- Identify a user that has the API Enabled permission and the Modify Metadata Through Metadata API Functions permission or Modify All Data permission. These permissions are required to access Metadata API calls.

  Note: If a user requires access to metadata but not to data, enable the Modify Metadata Through Metadata API Functions on page 7 permission. Otherwise, enable the Modify All Data permission.

- Install a SOAP client. Metadata API works with current SOAP development environments, including, but not limited to, Visual Studio® .NET and the Web Service Connector (WSC).

  In this document, we provide Java examples based on WSC and JDK 6 (Java Platform Standard Edition Development Kit 6). To run the samples, first download the latest force-wsc JAR file and its dependencies from mvnrepository.com/artifact/com.force.api/force-wsc/. Dependencies are listed on the page when you select a version.

  Note: Development platforms vary in their SOAP implementations. Implementation differences in certain development platforms can prevent access to some or all features in Metadata API.

Step 1: Generate or Obtain the Web Service WSDLs for Your Organization

To access Metadata API calls, you need a Web Service Description Language (WSDL) file. The WSDL file defines the Web service that is available to you. Your development platform uses this WSDL to generate stub code to access the Web service it defines. You can obtain the WSDL file from your organization's Salesforce administrator, or if you have access to the WSDL download page in the Salesforce user interface, you can generate it yourself. For more information about WSDL, see `http://www.w3.org/TR/wsdl`.

Before you can access Metadata API calls, you must authenticate to use the Web service using the `login()` call, which is defined in the enterprise WSDL and the partner WSDL. Therefore, you must also obtain one of these WSDLs.

Any user with the Modify Metadata Through Metadata API Functions or Modify All Data permission can download the WSDL file to integrate and extend the Salesforce platform.

Note: If a user requires access to metadata but not to data, enable the Modify Metadata Through Metadata API Functions on page 7 permission. Otherwise, enable the Modify All Data permission.

The sample code in Step 3: Walk Through the Java Sample Code on page 16 uses the enterprise WSDL, though the partner WSDL works equally well.

To generate the metadata and enterprise WSDL files for your organization:

1. Log in to your Salesforce account. You must log in as an administrator or as a user who has the "Modify All Data" permission.
2. From Setup, enter *API* in the `Quick Find` box, then select **API**.
3. Click **Generate Metadata WSDL**, and save the XML WSDL file to your file system.
4. Click **Generate Enterprise WSDL**, and save the XML WSDL file to your file system.

Step 2: Import the WSDL Files Into Your Development Platform

Once you have the WSDL files, import them into your development platform so that your development environment can generate the necessary objects for use in building client Web service applications. This section provides sample instructions for WSC. For instructions about other development platforms, see your platform's product documentation.

Note: The process for importing WSDL files is identical for the metadata and enterprise WSDL files.

Instructions for Java Environments (WSC)

Java environments access the API through Java objects that serve as proxies for their server-side counterparts. Before using the API, you must first generate these objects from your organization's WSDL file.

Each SOAP client has its own tool for this process. For WSC, use the `wsdlc` utility.

Note: Before you run `wsdlc`, you must have the WSC JAR file installed on your system and referenced in your classpath. You can download the latest force-wsc JAR file and its dependencies (dependencies are listed on the page when you select a version) from mvnrepository.com/artifact/com.force.api/force-wsc/.

The basic syntax for `wsdlc` is:

```
java -classpath pathToWsc;pathToWscDependencies com.sforce.ws.tools.wsdlc
pathToWsdl/WsdlFilename pathToOutputJar/OutputJarFilename
```

For example, on Windows:

```
java -classpath force-wsc-30.0.0.jar;ST4-4.0.7.jar;antlr-runtime-3.5.jar
com.sforce.ws.tools.wsdlc metadata.wsdl metadata.jar
```

On Mac OS X and Unix, use a colon instead of a semicolon in between items in the classpath:

```
java -classpath force-wsc-30.0.0.jar:ST4-4.0.7.jar:antlr-runtime-3.5.jar
com.sforce.ws.tools.wsdlc metadata.wsdl metadata.jar
```

`wsdlc` generates a JAR file and Java source code and bytecode files for use in creating client applications. Repeat this process for the enterprise WSDL to create an enterprise.JAR file.

Step 3: Walk Through the Java Sample Code

When you have imported the WSDL files, you can build client applications that use Metadata API. The sample is a good starting point for writing your own code.

Before you run the sample, modify your project and the code to:

1. Include the WSC JAR, its dependencies, and the JAR files you generated from the WSDLs.

   Note: Although WSC has other dependencies, the following sample only requires Rhino (`js-1.7R2.jar`), which you can download from mvnrepository.com/artifact/rhino/js.

2. Update USERNAME and PASSWORD variables in the `MetadataLoginUtil.login()` method with your user name and password. If your current IP address isn't in your organization's trusted IP range, you'll need to append a security token to the password.
3. If you are using a sandbox, be sure to change the login URL.

Login Utility

Java users can use `ConnectorConfig` to connect to Enterprise, Partner, and Metadata SOAP API. `MetadataLoginUtil` creates a `ConnectorConfig` object and logs in using the Enterprise WSDL login method. Then it retrieves `sessionId` and `metadataServerUrl` to create a `ConnectorConfig` and connects to Metadata API endpoint. `ConnectorConfig` is defined in WSC.

The `MetadataLoginUtil` class abstracts the login code from the other parts of the sample, allowing portions of this code to be reused without change across different Salesforce APIs.

```
import com.sforce.soap.enterprise.EnterpriseConnection;
import com.sforce.soap.enterprise.LoginResult;
import com.sforce.soap.metadata.MetadataConnection;
import com.sforce.ws.ConnectionException;
import com.sforce.ws.ConnectorConfig;

/**
 * Login utility.
 */
public class MetadataLoginUtil {

    public static MetadataConnection login() throws ConnectionException {
        final String USERNAME = "user@company.com";
        // This is only a sample. Hard coding passwords in source files is a bad practice.
```

```
        final String PASSWORD = "password";
        final String URL = "https://login.salesforce.com/services/Soap/c/64.0";
        final LoginResult loginResult = loginToSalesforce(USERNAME, PASSWORD, URL);
        return createMetadataConnection(loginResult);
    }

    private static MetadataConnection createMetadataConnection(
            final LoginResult loginResult) throws ConnectionException {
        final ConnectorConfig config = new ConnectorConfig();
        config.setServiceEndpoint(loginResult.getMetadataServerUrl());
        config.setSessionId(loginResult.getSessionId());
        return new MetadataConnection(config);
    }

    private static LoginResult loginToSalesforce(
            final String username,
            final String password,
            final String loginUrl) throws ConnectionException {
        final ConnectorConfig config = new ConnectorConfig();
        config.setAuthEndpoint(loginUrl);
        config.setServiceEndpoint(loginUrl);
        config.setManualLogin(true);
        return (new EnterpriseConnection(config)).login(username, password);
    }
}
```

Note: This example uses user and password authentication to obtain a session ID, which is then used for making calls to Metadata API. Alternatively, you can use OAuth authentication. After you athenticate with OAuth to Salesforce, pass the returned access token instead of the session ID. For example, pass the access token to the `setSessionId()` call on `ConnectorConfig`. To learn how to use OAuth authentication in Salesforce, see Authenticating Apps with OAuth in the Salesforce Help.

Java Sample Code for File-Based Development

The sample code logs in using the login utility. Then it displays a menu with retrieve, deploy, and exit.

The `retrieve()` and `deploy()` calls both operate on a .zip file named `components.zip`. The `retrieve()` call retrieves components from your organization into `components.zip`, and the `deploy()` call deploys the components in `components.zip` to your organization. If you save the sample to your computer and execute it, run the retrieve option first so that you have a `components.zip` file that you can subsequently deploy. After a retrieve call, the sample calls `checkRetrieveStatus()` in a loop until the operation is completed. Similarly, after a deploy call, the sample checks `checkDeployStatus()` in a loop until the operation is completed.

The `retrieve()` call uses a manifest file to determine the components to retrieve from your organization. A sample `package.xml` manifest file follows. For more details on the manifest file structure, see Deploying and Retrieving Metadata with the Zip File. For this sample, the manifest file retrieves all custom objects, custom tabs, and page layouts.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>CustomObject</name>
    </types>
    <types>
        <members>*</members>
```

```
        <name>CustomTab</name>
    </types>
    <types>
        <members>*</members>
        <name>Layout</name>
    </types>
    <version>64.0</version>
</Package>
```

Note the error handling code that follows each API call.

Note: This sample requires API version 34.0 or later.

```
import java.io.*;
import java.nio.channels.Channels;
import java.nio.channels.FileChannel;
import java.nio.channels.ReadableByteChannel;
import java.rmi.RemoteException;
import java.util.*;

import javax.xml.parsers.*;

import org.w3c.dom.*;
import org.xml.sax.SAXException;

import com.sforce.soap.metadata.*;

/**
 * Sample that logs in and shows a menu of retrieve and deploy metadata options.
 */
public class FileBasedDeployAndRetrieve {

    private MetadataConnection metadataConnection;

    private static final String ZIP_FILE = "components.zip";

    // manifest file that controls which components get retrieved
    private static final String MANIFEST_FILE = "package.xml";

    private static final double API_VERSION = 29.0;

    // one second in milliseconds
    private static final long ONE_SECOND = 1000;

    // maximum number of attempts to deploy the zip file
    private static final int MAX_NUM_POLL_REQUESTS = 50;

    private BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));


    public static void main(String[] args) throws Exception {
        FileBasedDeployAndRetrieve sample = new FileBasedDeployAndRetrieve();
        sample.run();
    }
```

```
public FileBasedDeployAndRetrieve() {
}

private void run() throws Exception {
    this.metadataConnection = MetadataLoginUtil.login();

    // Show the options to retrieve or deploy until user exits
    String choice = getUsersChoice();
    while (choice != null && !choice.equals("99")) {
        if (choice.equals("1")) {
            retrieveZip();
        } else if (choice.equals("2")) {
            deployZip();
        } else {
            break;
        }
        // show the options again
        choice = getUsersChoice();
    }
}

/*
 * Utility method to present options to retrieve or deploy.
 */
private String getUsersChoice() throws IOException {
    System.out.println(" 1: Retrieve");
    System.out.println(" 2: Deploy");
    System.out.println("99: Exit");
    System.out.println();
    System.out.print("Enter 1 to retrieve, 2 to deploy, or 99 to exit: ");
    // wait for the user input.
    String choice = reader.readLine();
    return choice != null ? choice.trim() : "";
}

private void deployZip() throws Exception {
    byte zipBytes[] = readZipFile();
    DeployOptions deployOptions = new DeployOptions();
    deployOptions.setPerformRetrieve(false);
    deployOptions.setRollbackOnError(true);
    AsyncResult asyncResult = metadataConnection.deploy(zipBytes, deployOptions);
    DeployResult result = waitForDeployCompletion(asyncResult.getId());
    if (!result.isSuccess()) {
        printErrors(result, "Final list of failures:\n");
        throw new Exception("The files were not successfully deployed");
    }
    System.out.println("The file " + ZIP_FILE + " was successfully deployed\n");
}

/*
* Read the zip file contents into a byte array.
*/
private byte[] readZipFile() throws Exception {
    byte[] result = null;
```

```
        // We assume here that you have a deploy.zip file.
        // See the retrieve sample for how to retrieve a zip file.
        File zipFile = new File(ZIP_FILE);
        if (!zipFile.exists() || !zipFile.isFile()) {
            throw new Exception("Cannot find the zip file for deploy() on path:"
                + zipFile.getAbsolutePath());
        }

        FileInputStream fileInputStream = new FileInputStream(zipFile);
        try {
            ByteArrayOutputStream bos = new ByteArrayOutputStream();
            byte[] buffer = new byte[4096];
            int bytesRead = 0;
            while (-1 != (bytesRead = fileInputStream.read(buffer))) {
                bos.write(buffer, 0, bytesRead);
            }

            result = bos.toByteArray();
        } finally {
            fileInputStream.close();
        }
        return result;
    }

    /*
    * Print out any errors, if any, related to the deploy.
    * @param result - DeployResult
    */
    private void printErrors(DeployResult result, String messageHeader) {
        DeployDetails details = result.getDetails();
        StringBuilder stringBuilder = new StringBuilder();
        if (details != null) {
            DeployMessage[] componentFailures = details.getComponentFailures();
            for (DeployMessage failure : componentFailures) {
                String loc = "(" + failure.getLineNumber() + ", " +
failure.getColumnNumber();
                if (loc.length() == 0 &&
!failure.getFileName().equals(failure.getFullName()))
                {
                    loc = "(" + failure.getFullName() + ")";
                }
                stringBuilder.append(failure.getFileName() + loc + ":"
                    + failure.getProblem()).append('\n');
            }
            RunTestsResult rtr = details.getRunTestResult();
            if (rtr.getFailures() != null) {
                for (RunTestFailure failure : rtr.getFailures()) {
                    String n = (failure.getNamespace() == null ? "" :
                        (failure.getNamespace() + ".")) + failure.getName();
                    stringBuilder.append("Test failure, method: " + n + "." +
                            failure.getMethodName() + " -- " + failure.getMessage() +
                            " stack " + failure.getStackTrace() + "\n\n");
                }
            }
```

```
            if (rtr.getCodeCoverageWarnings() != null) {
                for (CodeCoverageWarning ccw : rtr.getCodeCoverageWarnings()) {
                    stringBuilder.append("Code coverage issue");
                    if (ccw.getName() != null) {
                        String n = (ccw.getNamespace() == null ? "" :
                        (ccw.getNamespace() + ".")) + ccw.getName();
                        stringBuilder.append(", class: " + n);
                    }
                    stringBuilder.append(" -- " + ccw.getMessage() + "\n");
                }
            }
        }
        if (stringBuilder.length() > 0) {
            stringBuilder.insert(0, messageHeader);
            System.out.println(stringBuilder.toString());
        }
    }


    private void retrieveZip() throws Exception {
        RetrieveRequest retrieveRequest = new RetrieveRequest();
        // The version in package.xml overrides the version in RetrieveRequest
        retrieveRequest.setApiVersion(API_VERSION);
        setUnpackaged(retrieveRequest);

        AsyncResult asyncResult = metadataConnection.retrieve(retrieveRequest);
        RetrieveResult result = waitForRetrieveCompletion(asyncResult);

        if (result.getStatus() == RetrieveStatus.Failed) {
            throw new Exception(result.getErrorStatusCode() + " msg: " +
                    result.getErrorMessage());
        } else if (result.getStatus() == RetrieveStatus.Succeeded) {
         // Print out any warning messages
         StringBuilder stringBuilder = new StringBuilder();
         if (result.getMessages() != null) {
             for (RetrieveMessage rm : result.getMessages()) {
                 stringBuilder.append(rm.getFileName() + " - " + rm.getProblem() + "\n");

             }
         }
         if (stringBuilder.length() > 0) {
             System.out.println("Retrieve warnings:\n" + stringBuilder);
         }

         System.out.println("Writing results to zip file");
         File resultsFile = new File(ZIP_FILE);
         FileOutputStream os = new FileOutputStream(resultsFile);

         try {
             os.write(result.getZipFile());
         } finally {
             os.close();
         }
        }
```

```
    }

    private DeployResult waitForDeployCompletion(String asyncResultId) throws Exception {

        int poll = 0;
        long waitTimeMilliSecs = ONE_SECOND;
        DeployResult deployResult;
        boolean fetchDetails;
        do {
            Thread.sleep(waitTimeMilliSecs);
            // double the wait time for the next iteration

            waitTimeMilliSecs *= 2;
            if (poll++ > MAX_NUM_POLL_REQUESTS) {
                throw new Exception(
                    "Request timed out. If this is a large set of metadata components, "
+
                    "ensure that MAX_NUM_POLL_REQUESTS is sufficient.");
            }
            // Fetch in-progress details once for every 3 polls
            fetchDetails = (poll % 3 == 0);

          deployResult = metadataConnection.checkDeployStatus(asyncResultId, fetchDetails);

            System.out.println("Status is: " + deployResult.getStatus());
            if (!deployResult.isDone() && fetchDetails) {
                printErrors(deployResult, "Failures for deployment in progress:\n");
            }
        }
        while (!deployResult.isDone());

        if (!deployResult.isSuccess() && deployResult.getErrorStatusCode() != null) {
            throw new Exception(deployResult.getErrorStatusCode() + " msg: " +
                    deployResult.getErrorMessage());
        }

        if (!fetchDetails) {
            // Get the final result with details if we didn't do it in the last attempt.
            deployResult = metadataConnection.checkDeployStatus(asyncResultId, true);
        }

        return deployResult;
    }

    private RetrieveResult waitForRetrieveCompletion(AsyncResult asyncResult) throws
Exception {
     // Wait for the retrieve to complete
        int poll = 0;
        long waitTimeMilliSecs = ONE_SECOND;
        String asyncResultId = asyncResult.getId();
        RetrieveResult result = null;
        do {
            Thread.sleep(waitTimeMilliSecs);
            // Double the wait time for the next iteration
```

```
            waitTimeMilliSecs *= 2;
            if (poll++ > MAX_NUM_POLL_REQUESTS) {
                throw new Exception("Request timed out.  If this is a large set " +
                "of metadata components, check that the time allowed " +
                "by MAX_NUM_POLL_REQUESTS is sufficient.");
            }
            result = metadataConnection.checkRetrieveStatus(
                    asyncResultId, true);
            System.out.println("Retrieve Status: " + result.getStatus());
        } while (!result.isDone());

        return result;
    }

    private void setUnpackaged(RetrieveRequest request) throws Exception {
        // Edit the path, if necessary, if your package.xml file is located elsewhere
        File unpackedManifest = new File(MANIFEST_FILE);
        System.out.println("Manifest file: " + unpackedManifest.getAbsolutePath());

        if (!unpackedManifest.exists() || !unpackedManifest.isFile()) {
            throw new Exception("Should provide a valid retrieve manifest " +
                "for unpackaged content. Looking for " +
                unpackedManifest.getAbsolutePath());
        }

        // Note that we use the fully quualified class name because
        // of a collision with the java.lang.Package class
        com.sforce.soap.metadata.Package p = parsePackageManifest(unpackedManifest);
        request.setUnpackaged(p);
    }

    private com.sforce.soap.metadata.Package parsePackageManifest(File file)
            throws ParserConfigurationException, IOException, SAXException {
        com.sforce.soap.metadata.Package packageManifest = null;
        List<PackageTypeMembers> listPackageTypes = new ArrayList<PackageTypeMembers>();
        DocumentBuilder db =
                DocumentBuilderFactory.newInstance().newDocumentBuilder();
        InputStream inputStream = new FileInputStream(file);
        Element d = db.parse(inputStream).getDocumentElement();
        for (Node c = d.getFirstChild(); c != null; c = c.getNextSibling()) {
            if (c instanceof Element) {
                Element ce = (Element) c;
                NodeList nodeList = ce.getElementsByTagName("name");
                if (nodeList.getLength() == 0) {
                    continue;
                }
                String name = nodeList.item(0).getTextContent();
                NodeList m = ce.getElementsByTagName("members");
                List<String> members = new ArrayList<String>();
                for (int i = 0; i < m.getLength(); i++) {
                    Node mm = m.item(i);
                    members.add(mm.getTextContent());
                }
                PackageTypeMembers packageTypes = new PackageTypeMembers();
```

```
                packageTypes.setName(name);
                packageTypes.setMembers(members.toArray(new String[members.size()]));
                listPackageTypes.add(packageTypes);
            }
        }
        packageManifest = new com.sforce.soap.metadata.Package();
        PackageTypeMembers[] packageTypesArray =
                new PackageTypeMembers[listPackageTypes.size()];
        packageManifest.setTypes(listPackageTypes.toArray(packageTypesArray));
        packageManifest.setVersion(API_VERSION + "");
        return packageManifest;
    }
}
```

CHAPTER 4 Deploying and Retrieving Metadata

Use the deploy() and retrieve() calls to move metadata (XML files) between a Salesforce org and a local file system. After you retrieve your XML files into a file system, you can manage changes in a source-code control system, copy and paste code or setup configurations, diff changes to components, and perform many other file-based development operations. At any time you can deploy those changes to another Salesforce org.

Data in XML files is formatted using the English (United States) locale. This formatting ensures that fields that depend on locale, such as date fields, are interpreted consistently during data migrations between organizations using different languages. Organizations can support multiple languages for presentation to their users.

The deploy() and retrieve() calls are used primarily for these development scenarios:

- Development of a custom application (or customization) in a sandbox organization. After development and testing are completed, the application or customization is then deployed into a production organization using Metadata API.
- Team development of an application in a Developer Edition organization. After development and testing are completed, you can then distribute the application via Lightning Platform AppExchange.

You receive an API notification each time you retrieve 90% or more of the maximum number of custom fields that you can deploy at once with Metadata API. The maximum number of custom fields for one deployment is 45,000. The custom fields retrieved in one package.xml file are: 1) the sum of the fields on each object in the CustomObjects section of package.xml and 2) the sum of the custom fields in the CustomFields section of package.xml.

You can still retrieve above the deployable maximum up to the limit on total size of retrieved files. But you must use more than one deployment to deploy all of the custom fields.

Example: Warning: You've retrieved 47,000 instances of CustomField. You can't redeploy all these instances at the same time; the maximum is 45,000.

SEE ALSO:

Metadata Components and Types

Unsupported Metadata Types

Deploying and Retrieving Metadata with the Zip File

The `deploy()` and `retrieve()` calls are used to deploy and retrieve a .zip file. Within the .zip file is a project manifest `(package.xml)` that lists what to retrieve or deploy, and one or more XML components that are organized into folders.

Note: A component is an instance of a metadata type. For example, `CustomObject` is a metadata type for custom objects, and the `MyCustomObject__c` component is an instance of a custom object.

The files that are retrieved or deployed in a .zip file might be unpackaged components that reside in your org (such as *standard objects*) or packaged components that reside within named packages.

Note: You can deploy or retrieve up to 10,000 files at once. Managed packages use different limits: First-generation managed packages that have passed AppExchange Security Review can contain up to 35,000 files. Second-generation managed packages can contain up to 10,000 files. The maximum size of the deployed or retrieved .zip file is 39 MB. If the files are uncompressed in an unzipped folder, the size limit is 600 MB.

- If using the Ant Migration Tool to deploy an unzipped folder, all files in the folder are compressed first. The maximum size of uncompressed components in an unzipped folder is 600 MB or less depending on the compression ratio. If the files have a high compression ratio, you can migrate a total of approximately 600 MB because the compressed size would be under 39 MB. However, if the components can't be compressed much, like binary static resources, you can migrate less than 600 MB.
- Metadata API base-64 encodes components after they're compressed. The resulting .zip file can't exceed 50 MB, which is the limit for SOAP messages. Base-64 encoding increases the size of the payload, so your compressed payload can't exceed approximately 39 MB before encoding.
- You can perform a `retrieve()` call for a big object only if its index is defined. If a big object is created in Setup and doesn't yet have an index defined, you can't retrieve it.
- Limits can change without notice.

Every .zip file contains a project manifest, a file that's named `package.xml`, and a set of directories that contain the components. The manifest file defines the components that you're trying to retrieve or deploy in the .zip file. The manifest also defines the API version that's used for the deployment or retrieval.

Note: You can edit the project manifest, but be careful if you modify the list of components it contains. When you deploy or retrieve components, Metadata API references the components listed in the manifest, not the directories in the .zip file.

Note: Note: If you're retrieving any components that have dependencies by using the `rootTypesWithDependencies` parameter in the `RetrieveRequest` object, the dependent metadata components are added to the returned `.zip` file and `package.xml` file in the same directory as the root type that's being retrieved. This directory has a JSON file for each component with dependencies in the format `ComponentName.roottype.dependencies-meta.json`.

The following is a sample `package.xml` file. You can retrieve an individual component for a metadata type by specifying its `fullName` field value in a `members` element. You can also retrieve all components of a metadata type by using `<members>*</members>`.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>MyCustomObject__c</members>
        <name>CustomObject</name>
    </types>
    <types>
        <members>*</members>
        <name>CustomTab</name>
    </types>
    <types>
        <members>Standard</members>
        <name>Profile</name>
    </types>
    <version>64.0</version>
</Package>
```

The following elements can be defined in `package.xml`.

- `<fullName>` contains the name of the server-side package. If no `<fullName>` exists, the `package.xml` defines a client-side `unpackaged` package.

- `<types>` contains the name of the metadata type (for example, `CustomObject`) and the named members (for example, `myCustomObject__c`) to be retrieved or deployed. You can add multiple `<types>` elements in a manifest file.
- `<members>` contains the `fullName` of the component, for example `MyCustomObject__c`. The `listMetadata()` call is useful for determining the `fullName` for components of a particular metadata type if you want to retrieve an individual component. For many metadata types, you can replace the value in `members` with the wildcard character `*` (asterisk) instead of listing each member separately. See the reference topic for a specific type to determine whether that type supports wildcards.

  Note: You specify Security in the `<members>` element and Settings in the name element when retrieving the SecuritySettings component type.

- `<name>` contains the metadata type, for example `CustomObject` or `Profile`. There is one name defined for each metadata type in the directory. Any metadata type that extends Metadata is a valid value. The name that's entered must match a metadata type that's defined in the Metadata API WSDL. See Metadata Types for a list.
- `<version>` is the API version number that's used when the .zip file is deployed or retrieved. Currently the valid value is `64.0`.

For more sample `package.xml` manifest files that show you how to work with different subsets of metadata, see Sample `package.xml` Manifest Files.

To delete components, see Deleting Components from an Organization.

SEE ALSO:

Metadata Types

Slow Deployments

If a file-based Metadata API deployment occurs during server downtime, such as a Salesforce service upgrade, the deployment can take longer than expected. This behavior happens because both component deployment and validation are retried from the beginning after the service is restored. However, if Apex tests were part of the deployment, only tests that weren't run before the downtime are run.

This behavior affects file-based deployment and retrieval, change sets, some package installs and upgrades, second-generation managed package creation, and deploys and retrieves started from the Salesforce CLI or the Salesforce VS Code extensions. It doesn't affect CRUD-based metadata operations.

If your instance is due for a planned service upgrade, avoid running deployments during the service upgrade. To check whether your Salesforce instance is due for an upgrade, check Salesforce Trust. Salesforce performs major service upgrades three times per year and other maintenance updates throughout the year.

Does a Retrieve Job Have a Status of Pending?

If you initiate several concurrent retrieve operations for a single org, Metadata API automatically puts some of those jobs in a queue, if that becomes necessary for service protection. If a retrieve job has a status of `Pending`, it's in the queue. When one of the active retrieve jobs completes, Metadata API takes a pending job from the queue and activates it. If a retrieve job has a status of `InProgress`, it's active. The process repeats until the job queue is cleared.

For more information, see Metadata Limits in the *Salesforce Developer Limits and Allocations Quick Reference*.

Sample `package.xml` Manifest Files

This section includes sample package.xml manifest files that show you how to work with different subsets of metadata. A manifest file can include multiple <types> elements so you could combine the individual samples into one package.xml manifest file if you want to work with all the metadata in one batch.

The following samples are listed:

- Standard Objects on page 28
- All Custom Objects on page 28
- Standard Picklist Fields on page 29
- Custom and Standard Fields on page 30
- List Views for Standard Objects on page 30
- Packages on page 31
- Security Settings on page 31
- Assignment Rules, Auto-Response Rules, Escalation Rules on page 32
- Sharing Rules on page 32
- Managed Component Access on page 33

For more information about the structure of a manifest file, see Deploying and Retrieving Metadata with the Zip File.

Standard Objects

This sample `package.xml` manifest file illustrates how to work with the standard Account object. Retrieving or deploying a standard object includes all custom and standard fields except for standard fields that aren't customizable. All custom fields are supported. Only standard fields that you can customize are supported, that is, standard fields to which you can add help text or enable history tracking or Chatter feed tracking. Other standard fields aren't supported, including system fields (such as `CreatedById` or `LastModifiedDate`) and autonumber fields.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>Account</members>
        <name>CustomObject</name>
    </types>
    <version>64.0</version>
</Package>
```

Note how you work with the standard Account object by specifying it as a member of a CustomObject type. However, you can't use an asterisk wildcard to work with all standard objects; each standard object must be specified by name.

All Custom Objects

This sample `package.xml` manifest file illustrates how to work with all custom objects.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>CustomObject</name>
```

```
    </types>
    <version>64.0</version>
</Package>
```

This manifest file can be used to retrieve or deploy all custom objects, but not all standard objects.

Standard Picklist Fields

In API version 38.0 and later, the StandardValueSet type represents standard picklists. Picklists are no longer represented by fields as in earlier versions. This sample `package.xml` represents the `Industry` standard picklist as a StandardValueSet type.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>Industry</members>
        <name>StandardValueSet</name>
    </types>
    <version>64.0</version>
</Package>
```

Note: The name of a standard value set is case-sensitive.

The `Industry` standard value set corresponds to the `Account.Industry` or `Lead.Industry` field in API version 37.0 and earlier. This example shows a `package.xml` sample for the `Account.Industry` picklist.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>Account.Industry</members>
        <name>CustomField</name>
    </types>
    <version>37.0</version>
</Package>
```

Note: The name of a picklist field is case-sensitive.

Note the ***objectName.picklistField*** syntax in the `<members>` field where *`objectName`* is the name of the object, such as `Account`, and *`picklistField`* is the name of the standard picklist field, such as `Industry`.

This next `package.xml` sample represents opportunity team roles in API version 38.0 and later. Specify opportunity team roles as a `SalesTeamRole` standard value set. Opportunity team roles have the same picklist values as the account team roles.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>SalesTeamRole</members>
        <name>StandardValueSet</name>
    </types>
    <version>64.0</version>
</Package>
```

The `SalesTeamRole` standard value set corresponds to one of these field names in API version 37.0 and earlier: `OpportunityTeamMember.TeamMemberRole`, `UserAccountTeamMember.TeamMemberRole`,

`UserTeamMember.TeamMemberRole`, and `AccountTeamMember.TeamMemberRole`. Opportunity team roles are represented in this sample `package.xml` as the `OpportunityTeamMember.TeamMemberRole` field.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>OpportunityTeamMember.TeamMemberRole</members>
        <name>CustomField</name>
    </types>
    <version>37.0</version>
</Package>
```

To learn about the names of standard value sets and how they map to picklist field names, see StandardValueSet Names and Standard Picklist Fields.

Custom and Standard Fields

This sample `package.xml` manifest file illustrates how to work with custom fields in custom and standard objects and standard fields in a standard object.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>MyCustomObject__c.MyCustomField__c</members>
        <name>CustomField</name>
    </types>
    <types>
        <members>Account.SLA__c</members>
        <members>Account.Phone</members>
        <name>CustomField</name>
    </types>
    <version>64.0</version>
</Package>
```

Note the ***objectName.field*** syntax in the `<members>` field where *objectName* is the name of the object, such as Account, and *field* is the name of the custom or standard field, such as an `SLA` picklist field representing a service-level agreement option. The `MyCustomField` custom field in the MyCustomObject custom object is uniquely identified by its full name `MyCustomObject__c.MyCustomField__c`. Similarly, the `Phone` standard field in the Account standard object is uniquely identified by its full name `Account.Phone`.

All custom fields are supported. Only standard fields that you can customize are supported, that is, standard fields to which you can add help text or enable history tracking or Chatter feed tracking. Other standard fields aren't supported, including system fields (such as `CreatedById` or `LastModifiedDate`) and autonumber fields.

List Views for Standard Objects

The easiest way to retrieve list views for a standard object is to retrieve the object. The list views are included in the retrieved component. See the section of this topic on Standard Objects.

You can also work with individual list views if you don't want to retrieve all the details for the object. This sample `package.xml` manifest file illustrates how to work with a list view for the standard Account object.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
```

```
    <types>
        <members>Account.AccountTeam</members>
        <name>ListView</name>
    </types>
    <version>64.0</version>
</Package>
```

Note the ***objectName.listViewUniqueName*** syntax in the `<members>` field where *objectName* is the name of the object, such as Account, and *listViewUniqueName* is the `View Unique Name` for the list view. If you retrieve this list view, the component is stored in `objects/Account.object`.

Packages

To retrieve a package, set the name of the package in the `packageNames` field in RetrieveRequest when you call `retrieve()`. The `package.xml` manifest file is automatically populated in the retrieved `.zip` file. The `<fullName>` element in `package.xml` contains the name of the retrieved package.

If you use an asterisk wildcard in a `<members>` element to retrieve all the components of a particular metadata type, the retrieved contents don't include components in managed packages.

For more information about managed packages, see the Second-Generation Managed Packaging Developer Guide.

The easiest way to retrieve a component in a managed package is to retrieve the complete package by setting the name of the package in the `packageNames` field in `RetrieveRequest`, as described earlier. The following sample `package.xml` manifest file illustrates an alternative to retrieve an individual component in a package.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>myns__MyCustomObject__c</members>
        <name>CustomObject</name>
    </types>
    <version>64.0</version>
</Package>
```

Note the ***namespacePrefix*__*objectName*** syntax in the `<members>` field where *namespacePrefix* is the namespace prefix of the package and *objectName* is the name of the object. A namespace prefix is a 1-character to 15-character alphanumeric identifier that distinguishes your package and its contents from other publishers' packages. For more information, see Create and Register Your Namespace for Second-Generation Managed Packages.

Note: The namespace prefix is important to help identify the source of items like fields, custom objects, and more from different managed packages. For example, when working with FlexiPages in your org, we recommend against removing namespaces for object fields, because it can cause unexpected results such as name collisions.

Security Settings

This sample `package.xml` manifest file illustrates how to work with an org's security settings. You specify Security in the `<members>` element and Settings in the name element when retrieving the SecuritySettings component type.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>Security</members>
        <name>Settings</name>
```

```
    </types>
    <version>64.0</version>
</Package>
```

Assignment Rules, Auto-Response Rules, Escalation Rules

Assignment rules, auto-response rules, and escalation rules use different `package.xml` type names to access sets of rules or individual rules for object types. For example, this sample `package.xml` manifest file illustrates how to access an org's assignment rules for just Cases and Leads.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>Case</members>
        <members>Lead</members>
        <name>AssignmentRules</name>
    </types>
    <version>64.0</version>
</Package>
```

The following sample `package.xml` manifest file illustrates how to access just the "samplerule" Case assignment rule and the "newrule" Lead assignment rule. Notice that the type name is `AssignmentRule` and not `AssignmentRules`.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>Case.samplerule</members>
        <members>Lead.newrule</members>
        <name>AssignmentRule</name>
    </types>
    <version>64.0</version>
</Package>
```

Similarly, for accessing individual auto-response rules and escalation rules, use `AutoResponseRule` and `EscalationRule` instead of `AutoResponseRules` and `EscalationRules`.

Sharing Rules

In API version 33.0 and later, you can retrieve and deploy sharing rules for all standard and custom objects. This sample `package.xml` manifest file illustrates how to work with an org's sharing rules, such as retrieving a specific criteria-based sharing rule for the lead object, retrieving all ownership-based sharing rules for all objects, and retrieving all territory-based sharing rules for the account object.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>Lead.testShareRule</members>
         <name>SharingCriteriaRule</name>
    </types>
    <types>
        <members>*</members>
        <name>SharingOwnerRule</name>
    </types>
    <types>
```

```
        <members>Account.*</members>
        <name>SharingTerritoryRule</name>
    </types>
    <version>33.0</version>
</Package>
```

Managed Component Access

In API version 29.0 and later, you can retrieve and deploy access settings for these managed components in profiles and permission sets:

- Apex classes
- Apps
- Custom field permissions
- Custom object permissions
- Custom tab settings
- External data sources
- Record types
- Visualforce pages

In API version 51.0 and later, you can retrieve and deploy access settings for login flows.

When retrieving and deploying managed component permissions, specify the namespace followed by two underscores. Wildcards aren't supported.

For example, let's say you install a managed package with the namespace `MyNamespace` and the custom object `JobRequest__c`. To set object permissions for `JobRequest__c` in the package to the custom profile MyProfile, you would add the following to the `.profile` file.

To deploy:

```
<objectPermissions>
    <allowCreate>true</allowCreate>
    <allowDelete>true</allowDelete>
    <allowEdit>true</allowEdit>
    <allowRead>true</allowRead>
    <viewAllRecords>false</viewAllRecords>
    <modifyAllRecords>false</modifyAllRecords>
    <object>MyNamespace__JobRequest__c</object>
</objectPermissions>
```

To retrieve:

```
<types>
    <members>MyNamespace__JobRequest__c</members>
    <name>CustomObject</name>
</types>
<types>
    <members>MyProfile</members>
    <name>Profile</name>
</types>
```

When retrieving permission sets and profiles, make sure that you also retrieve any components that are related to the permissions and settings. For example, when retrieving app visibilities, you must also retrieve the associated app, and when retrieving object or field permissions, you must also retrieve the associated object.

Running Tests in a Deployment

Default Test Execution in Production

When no test level is specified in the deployment options, the default test execution behavior depends on the contents of your deployment package. When deploying to production, all tests, except those that originate from managed packages, are executed if your deployment package contains Apex classes or triggers. If your package doesn't contain Apex components, no tests are run by default.

In API version 33.0 and earlier, tests were run for components that required tests, such as custom objects, and not only for Apex components. For example, if your package contains a custom object, all tests are run in API version 33.0 and earlier. In contrast, starting with API version 34.0, no tests are run for this package. The API version corresponds to the version of your API client or the version of the tool you're using (Ant Migration Tool).

You can run tests for a deployment of non-Apex components. You can override the default test execution behavior by setting the test level in your deployment options. Test levels are enforced regardless of the types of components present in your deployment package. We recommend that you run all local tests in your development environment, such as sandbox, before deploying to production. Running tests in your development environment reduces the number of tests needed to run in a production deployment.

Default Test Execution in Production for API Version 33.0 and Earlier

For deployment to a production organization, all local tests in your organization are run by default. Tests that originate from installed managed packages aren't run by default. If any test fails, the entire deployment is rolled back.

If the deployment includes components for the following metadata types, all local tests are run.

- ApexClass
- ApexComponent
- ApexPage
- ApexTrigger
- ArticleType
- BaseSharingRule
- CriteriaBasedSharingRule
- CustomField
- CustomObject
- DataCategoryGroup
- Flow
- InstalledPackage
- NamedFilter
- OwnerSharingRule
- PermissionSet
- Profile
- Queue
- RecordType
- RemoteSiteSetting
- Role
- SharingReason

- Territory
- Validation Rules
- Workflow

For example, no tests are run for the following deployments:

- 1 CustomApplication component
- 100 Report components and 40 Dashboard components

But all local tests are run for any of the following example deployments, because they include at least one component from the list above:

- 1 CustomField component
- 1 ApexComponent component and 1 ApexClass component
- 5 CustomField components and 1 ApexPage component
- 100 Report components, 40 Dashboard components, and 1 CustomField component

SEE ALSO:

deploy()

Running a Subset of Tests in a Deployment

Test levels enable you to have more control over which tests are run in a deployment. To shorten deployment time to production, run a subset of tests when deploying Apex components. The default test execution behavior in production has also changed. By default, if no test level is specified, no tests are executed, unless your deployment package contains Apex classes or triggers.

If the code coverage of an Apex component in the deployment is less than 75%, the deployment fails. If one of the specified tests fails, the deployment also fails. We recommend that you test your deployment in sandbox first to ensure that the specified tests cover each component sufficiently. Even if your organization's overall code coverage is 75% or more, the individual coverage of the Apex components being deployed can be less. If the code coverage requirement isn't met, write more tests and include them in the deployment.

To run a subset of tests, set the `RunSpecifiedTests` test level on the `DeployOptions` object. Next, specify each test class to run in `DeployOptions`. Finally, pass `DeployOptions` as an argument to the `deploy()` call. The following example performs those steps to run only the specified test classes.

```
// Create the DeployOptions object.
DeployOptions deployOptions = new DeployOptions();

// Set the appropriate test level.
deployOptions.setTestLevel(TestLevel.RunSpecifiedTests);

// Specify the test classes to run.
// String array contains test class names.
String[] tests = {"TestClass1", "TestClass2", "TestClass3"};
// Add the test class names array to the deployment options.
deployOptions.setRunTests(tests);

// Call deploy() by passing the deployment options object as an argument.
AsyncResult asyncResult = metadatabinding.deploy(zipBytes,deployOptions);
```

Notes About Running Specific Tests

- You can specify only test classes. You can't specify individual test methods.
- We recommend that you refactor test classes to include the minimum number of tests that meet code coverage requirements. Refactoring your test classes can contribute to shorter test execution times, and as a result, shorter deployment times.
- You can deactivate a trigger in the target organization by deploying it with an inactive state. However, the trigger must have been previously deployed with an active state.

Run the Same Tests in Sandbox and Production Deployments

Starting in API version 34.0, you can choose which tests to run in your development environment, such as only local tests, to match the tests run in production. In earlier versions, if you enabled tests in your sandbox deployment, you couldn't exclude managed package tests.

By default, no tests are run in a deployment to a non-production organization, such as a sandbox or a Developer Edition organization. To specify tests to run in your development environment, set a `testLevel` deployment option. For example, to run local tests in a deployment and to exclude managed package tests, set `testLevel` on the `DeployOptions` object to `TestLevel.RunLocalTests`. Next, pass this object as an argument to the `deploy()` call as follows.

```
// Create the DeployOptions object.
DeployOptions deployOptions = new DeployOptions();

// Set the appropriate test level.
deployOptions.setTestLevel(TestLevel.RunLocalTests);

// Call deploy() by passing the deployment options object as an argument.
AsyncResult asyncResult = metadatabinding.deploy(zipBytes,deployOptions);
```

Note: The `RunLocalTests` test level is enforced regardless of the contents of the deployment package. In contrast, tests are executed by default in production only if your deployment package contains Apex classes or triggers. You can use `RunLocalTests` for sandbox and production deployments.

Limit on Enqueued Deployments from Apex

We limit the number of Metadata API deployments originating from Apex that can be enqueued at a time. This limit helps preserve service function and resources for all customers on a server. Because this limit is a queue-depth limit, as long as the server can keep dequeuing, you can keep enqueuing deploys through Apex. This limit is based on analysis to make sure that it doesn't affect your day-to-day operations.

When you reach the limit, you receive this exception as an API response in Apex.

```
[
{
  "message" : "The service received too many metadata deployment requests from Apex and
doesn't have the resources to accept new requests",
  "errorCode" : "System.AsyncException"
}
]
```

The limit applies only to enqueued Metadata API deployments that originate from Apex. It doesn't affect Metadata API deployments from Salesforce CLI, change sets, or packaging. The limit does apply if a package contains Apex that triggers metadata deployments. It also applies to the Metadata.Operations.enqueueDeployment() Apex method. This limit applies to all Salesforce editions.

Maintaining User References

User fields are preserved during a metadata deployment.

When a component in your deployment refers to a specific user, such as a recipient of a workflow email notification or a dashboard running user, then Salesforce attempts to locate a matching user in the destination organization by comparing usernames during the deployment.

For example, when you copy data to a sandbox, the fields containing usernames from the production organization are altered to include the sandbox name. In a sandbox named `test`, the username `user@acme.com` becomes `user@acme.com.test`. When you deploy the metadata in the sandbox to another organization, the `test` in the username is ignored.

For user references in deployments, Salesforce performs the following sequence:

1. Salesforce compares usernames in the source environment to the destination environment and adapts the organization domain name.
2. If two or more usernames match, Salesforce lists the matching names and requests one of the users in the source environment be renamed.
3. If a username in the source environment doesn't exist in the destination environment, Salesforce displays an error, and the deployment stops until the usernames are removed or resolved to users in the destination environment.

CHAPTER 5 CRUD-Based Metadata Development

Use the CRUD-based metadata calls to create, update, or delete setup and configuration components for your organization or application. These configuration components include custom objects, custom fields, and other configuration metadata. The metadata calls mimic the behavior in the Salesforce user interface for creating, updating, or deleting components. Whatever rules apply there also apply to these calls.

Metadata calls are different from the core, synchronous API calls in these ways.

- Metadata API calls are available in a separate WSDL. To download the WSDL, log into Salesforce, from Setup, enter `API` in the `Quick Find` box, then select **API** and click the Download Metadata WSDL link.
- After logging in, you must send Metadata API calls to the Metadata API endpoint, which has a different URL than SOAP API. Retrieve the `metadataServerUrl` from the LoginResult returned by your SOAP API `login()` call. For more information about SOAP API, see the *SOAP API Developer Guide*.
- Metadata calls are either synchronous or asynchronous. CRUD calls are synchronous in API version 30.0 and later, and similar to the API core calls the results are returned in a single call. In earlier API versions, create, update, and delete are only asynchronous, which means that the results aren't immediately returned in one call.
- There are synchronous metadata calls that map to the corresponding core SOAP API synchronous calls.
  - createMetadata() maps to the `create()` SOAP API call.
  - updateMetadata() maps to the `update()` SOAP API call.
  - deleteMetadata() maps to the `delete()` SOAP API call.

Note: Metadata API also supports `retrieve()` and `deploy()` calls for retrieving and deploying metadata components. For more information, see Deploying and Retrieving Metadata.

Java Sample for CRUD-Based Development with Synchronous Calls

This section guides you through a sample Java client application that uses CRUD-based calls. This sample application performs the following main tasks.

1. Uses the `MetadataLoginUtil.java` class to create a Metadata connection. For more information, see Step 3: Walk Through the Java Sample Code.
2. Calls createMetadata() to create a custom object. This call returns the result in one call.
3. Inspects the returned `SaveResult` object to check if the operation succeeded, and if it didn't, writes the component name, error message, and status code to the output.

```
import com.sforce.soap.metadata.*;

/**
 * Sample that logs in and creates a custom object through the metadata API
 */
public class CRUDSampleCreate {
```

```
    private MetadataConnection metadataConnection;

    // one second in milliseconds
    private static final long ONE_SECOND = 1000;

    public CRUDSampleCreate() {
    }

    public static void main(String[] args) throws Exception {
        CRUDSampleCreate crudSample = new CRUDSampleCreate();
        crudSample.runCreate();
    }

    /**
     * Create a custom object. This method demonstrates usage of the
     * create() and checkStatus() calls.
     *
     * @param uniqueName Custom object name should be unique.
     */
    private void createCustomObjectSync(final String uniqueName) throws Exception {
        final String label = "My Custom Object";
        CustomObject co = new CustomObject();
        co.setFullName(uniqueName);
        co.setDeploymentStatus(DeploymentStatus.Deployed);
        co.setDescription("Created by the Metadata API Sample");
        co.setEnableActivities(true);
        co.setLabel(label);
        co.setPluralLabel(label + "s");
        co.setSharingModel(SharingModel.ReadWrite);

        // The name field appears in page layouts, related lists, and elsewhere.
        CustomField nf = new CustomField();
        nf.setType(FieldType.Text);
        nf.setDescription("The custom object identifier on page layouts, related lists
etc");
        nf.setLabel(label);
        nf.setFullName(uniqueName);
        customObject.setNameField(nf);

        SaveResult[] results = metadataConnection
                .createMetadata(new Metadata[] { co });

        for (SaveResult r : results) {
            if (r.isSuccess()) {
                System.out.println("Created component: " + r.getFullName());
            } else {
                System.out
                        .println("Errors were encountered while creating "
                                + r.getFullName());
                for (Error e : r.getErrors()) {
                    System.out.println("Error message: " + e.getMessage());
                    System.out.println("Status code: " + e.getStatusCode());
                }
            }
```

```
        }
    }

    private void runCreate() throws Exception {
        metadataConnection = MetadataLoginUtil.login();
        // Custom objects and fields must have __c suffix in the full name.
        final String uniqueObjectName = "MyCustomObject__c";
        createCustomObjectSync(uniqueObjectName);
    }
}
```

Java Sample for CRUD-Based Development with Asynchronous Calls

Important: The sample in this section depends on the asynchronous `create()` CRUD call. Asynchronous CRUD calls are no longer available as of API version 31.0 and are available only in earlier API versions.

This section guides you through a sample Java client application that uses asynchronous CRUD-based calls. This sample application performs the following main tasks:

1. Uses the `MetadataLoginUtil.java` class to create a Metadata connection. For more information, see Step 3: Walk Through the Java Sample Code.
2. Calls create() to create a custom object.

   Salesforce returns an AsyncResult object for each component you tried to create. The AsyncResult object is updated with status information as the operation moves from a queue to completed or error state.
3. Calls checkStatus() in a loop until the status value in AsyncResult indicates that the create operation is completed.

Note the error handling code that follows each API call.

```
import com.sforce.soap.metadata.*;

/**
 * Sample that logs in and creates a custom object through the metadata api
 */
public class CRUDSample {
    private MetadataConnection metadataConnection;

    // one second in milliseconds
    private static final long ONE_SECOND = 1000;

    public CRUDSample() {
    }

    public static void main(String[] args) throws Exception {
        CRUDSample crudSample = new CRUDSample();
        crudSample.runCreate();
    }

    /**
     * Create a custom object. This method demonstrates usage of the
     * create() and checkStatus() calls.
     *
     * @param uniqueName Custom object name should be unique.
```

```
     */
    private void createCustomObject(final String uniqueName) throws Exception {
        final String label = "My Custom Object";
        CustomObject customObject = new CustomObject();
        customObject.setFullName(uniqueName);
        customObject.setDeploymentStatus(DeploymentStatus.Deployed);
        customObject.setDescription("Created by the Metadata API Sample");
        customObject.setLabel(label);
        customObject.setPluralLabel(label + "s");
        customObject.setSharingModel(SharingModel.ReadWrite);

        // The name field appears in page layouts, related lists, and elsewhere.
        CustomField nf = new CustomField();
        nf.setType(FieldType.Text);
        nf.setDescription("The custom object identifier on page layouts, related lists
etc");
        nf.setLabel(label);
        nf.setFullName(uniqueName);
        customObject.setNameField(nf);

        AsyncResult[] asyncResults = metadataConnection.create(
            new CustomObject[]{customObject});
        if (asyncResults == null) {
            System.out.println("The object was not created successfully");
            return;
        }

        long waitTimeMilliSecs = ONE_SECOND;

       // After the create() call completes, we must poll the results of the checkStatus()

        // call until it indicates that the create operation has completed.
        do {
            printAsyncResultStatus(asyncResults);
            waitTimeMilliSecs *= 2;
            Thread.sleep(waitTimeMilliSecs);
            asyncResults = metadataConnection.checkStatus(new
String[]{asyncResults[0].getId()});
        } while (!asyncResults[0].isDone());

        printAsyncResultStatus(asyncResults);
    }

    private void printAsyncResultStatus(AsyncResult[] asyncResults) throws Exception {
        if (asyncResults == null || asyncResults.length == 0 || asyncResults[0] == null)
{
            throw new Exception("The object status cannot be retrieved");
        }

       AsyncResult asyncResult = asyncResults[0]; //we are creating only 1 metadata object


        if (asyncResult.getStatusCode() != null) {
            System.out.println("Error status code: " +
```

```
                    asyncResult.getStatusCode());
            System.out.println("Error message: " + asyncResult.getMessage());
        }

        System.out.println("Object with id:" + asyncResult.getId() + " is " +
            asyncResult.getState());
    }

    private void runCreate() throws Exception {
        metadataConnection = MetadataLoginUtil.login();
        // Custom objects and fields must have __c suffix in the full name.
        final String uniqueObjectName = "MyCustomObject__c";
        createCustomObject(uniqueObjectName);
    }
}
```

CHAPTER 6 REST Resources

In this chapter ...

- Deploy Metadata with Apex Testing Using REST
- Deploy Metadata with REST API in Salesforce CLI

Use the REST resource `deployRequest` to move metadata (XML files) between a Salesforce organization and a local file system.

Data in XML files is formatted using the English (United States) locale. This approach ensures that fields that depend on locale, such as date fields, are interpreted consistently during data migrations between organizations using different languages. Organizations can support multiple languages for presentation to their users.

Metadata deployment is used primarily for the following development scenarios.

- Development of a custom application (or customization) in a sandbox organization. After development and testing are completed, the application or customization is then deployed into a production organization using Metadata API.
- Team development of an application in a Developer Edition organization. After development and testing are completed, you can then distribute the application via Lightning Platform AppExchange.

Working with the Zip File

The `deployRequest` resource is used to deploy a .zip file. Within the .zip file is a project manifest `(package.xml)` that lists what to retrieve or deploy, and one or more XML components that are organized into folders.

Note: A component is an instance of a metadata type. For example, `CustomObject` is a metadata type for custom objects, and the `MyCustomObject__c` component is an instance of a custom object.

The files that are deployed in a `.zip` file can be unpackaged components that reside in your organization (such as *standard objects*). The files can also be packaged components that reside within named packages.

Note: You can deploy up to 10,000 files at once. (In API version 43.0 and later, AppExchange packages can contain up to 12,500 files.) The .zip file size limit that applies to SOAP calls doesn't apply to the `deployRequest` REST resource. However, the 400-MB limit for components that are uncompressed into an unzipped folder after upload applies to both SOAP and REST deployments.

Every .zip file contains a project manifest, a file that's named `package.xml`, and a set of directories that contain the components. The manifest file defines the components that you're trying to retrieve or deploy and the API version used for the deployment or retrieval.

The following is a sample `package.xml` file.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
```

```
        <members>MyCustomObject__c</members>
        <name>CustomObject</name>
    </types>
    <types>
        <members>*</members>
        <name>CustomTab</name>
    </types>
    <types>
        <members>Standard</members>
        <name>Profile</name>
    </types>
    <version>64.0</version>
</Package>
```

The following elements can be defined in `package.xml`.

- `<fullName>` contains the name of the server-side package. If no `<fullName>` exists, it's a client-side `unpackaged` package.
- `<types>` contains the name of the metadata type (for example, `CustomObject`) and the named members (for example, `myCustomObject__c`) to be deployed. You can add multiple `<types>` elements in a manifest file.
- `<members>` contains the `fullName` of the component, such as `MyCustomObject__c`. For many metadata types, you can replace the value in `members` with the wildcard character `*` (asterisk) instead of listing each member separately. For a list of metadata types that allow the wildcard character, see the "Allows Wildcard (*)?" column in Metadata Types.

  Note: You specify Security in the `<members>` element and Settings in the name element when retrieving the SecuritySettings component type.

- `<name>` contains the metadata type, for example `CustomObject` or `Profile`. There's one name defined for each metadata type in the directory. Any metadata type that extends Metadata is a valid value. The name that's entered must match a metadata type that's defined in the Metadata API WSDL. See Metadata Types for a list.
- `<version>` is the API version number that's used when the .zip file is deployed or retrieved. Currently the valid value is `64.0`.

For more sample `package.xml` manifest files that show you how to work with different subsets of metadata, see Sample package.xml Manifest Files.

To delete components, see Deleting Components from an Organization.

Deploy Metadata with Apex Testing Using REST

Deploy using the `deployRequest` REST resource to initiate a request that handles all operations for the deployment.

You can deploy or retrieve up to 10,000 files at once. Managed packages use different limits: First-generation managed packages that have passed AppExchange Security Review can contain up to 35,000 files. Second-generation managed packages can contain up to 10,000 files. The maximum size of the deployed or retrieved .zip file is 39 MB. If the files are uncompressed in an unzipped folder, the size limit is 600 MB.

URI

`https://`***`host`***`/services/data/v`***`XX.`***`0/metadata/deployRequest`

Formats

JSON

HTTP Method

POST

Authentication

`Authorization: Bearer` ***`token`***

deployOptions Parameters

Note: To review the default testing behavior for deployments and approaches that can save time while still enabling you to meet testing requirements, see Running Tests in a Deployment and Run the Same Tests in Sandbox and Production Deployments.

Parameter	Description
`allowMissingFiles`	Boolean. If files that are specified in `package.xml` aren't in the `.zip` file, specifies whether a deployment can still succeed. Don't set this argument for deployment to production orgs.
`autoUpdatePackage`	Reserved for future use.
`checkOnly`	Boolean. Defaults to `false`. Set to `true` to perform a test deployment (validation) of components without saving the components in the target org. A validation enables you to verify the results of tests that would be generated in a deployment, but doesn't commit any changes. After a validation finishes with passing tests, it can qualify for deployment without rerunning tests. See Deploy a Recently Validated Component Set Without Tests.
`ignoreWarnings`	Boolean. Indicates whether a deployment is allowed to complete successfully despite one or more warnings (`true`) or not (`false`). Defaults to `false`. The DeployMessage object for a warning contains the following values: • `problemType`—`Warning` • `problem`—The text of the warning. If a warning occurs and `ignoreWarnings` is set to `true`, the `success` field in DeployMessage is `true`. If `ignoreWarnings` is set to `false`, `success` is set to `false` and the warning is treated like an error.
`performRetrieve`	Reserved for future use.

Parameter	Description
`purgeOnDelete`	Boolean. If `true`, the deleted components in the `destructiveChanges.xml` manifest file aren't stored in the Recycle Bin. Instead, they become immediately eligible for deletion. This option only works in Developer Edition or sandbox orgs. It doesn't work in production orgs.
`rollbackOnError`	Boolean. Indicates whether any failure causes a complete rollback (`true`) or not (`false`). If `false`, whatever actions can be performed without errors are performed, and errors are returned for the remaining actions. This parameter must be set to `true` if you're deploying to a production org. The default is `false`.
`runTests`	String[]. A list of Apex tests to run during deployment. Specify the class name, one name per instance. The class name can also specify a namespace with a dot notation. For more information, see Running a Subset of Tests in a Deployment. To use this option, set `testLevel` to `RunSpecifiedTests`.
`singlePackage`	Boolean. Indicates whether the specified `.zip` file points to a directory structure with a single package (`true`) or a set of packages (`false`).
`testLevel`	TestLevel (enumeration of type string). Optional. Specifies which tests are run as part of a deployment. The test level is enforced regardless of the types of components that are present in the deployment package. Valid values are: • `NoTestRun`—No tests are run. This test level applies only to deployments to development environments, such as sandbox, Developer Edition, or trial organizations. This test level is the default for development environments. • `RunSpecifiedTests`—Only the tests that you specify in the `runTests` option are run. Code coverage requirements differ from the default coverage requirements when using this test level. Each class and trigger in the deployment package must be covered by the executed tests for a minimum of 75% code coverage. This coverage is computed for each class and triggers individually and is different than the overall coverage percentage. • `RunLocalTests`—All tests in your org are run, except the ones that originate from installed managed and unlocked packages. This test level is the default for production deployments that include Apex classes or triggers. • `RunAllTestsInOrg`—All tests are run. The tests include all tests in your org, including tests of managed packages. If you don't specify a test level, the default test execution behavior is used. See Running Tests in a Deployment. Apex tests that run as part of a deployment always run synchronously and serially.

Request Body: Deploy Metadata

When you deploy metadata, your request includes both the deployment parameters and the .zip file containing the component directories and the manifest. Set the header to `Content-Type: multipart/form-data`,

This example POST request creates a `deployRequest` object that initiates a deployment.

1. The POST request header is set to `Content-Type: multipart/form-data` and defines a `boundary` value to encapsulate different subparts of the request.
2. In the subpart after the first boundary, a JSON request creates a `deployOptions` child object for passing the deployment parameters.
3. The subpart after the second boundary specifies the .zip file containing the manifest and the component directories.

```
POST /services/data/v48.0/metadata/deployRequest
Authorization: Bearer 00D....
Content-Type: multipart/form-data; boundary=----------------------------BOUNDARY
----------------------------BOUNDARY
Content-Disposition: form-data; name="json"
Content-Type: application/json
{
    "deployOptions" :
        {
        "allowMissingFiles" : false,
        "autoUpdatePackage" : false,
        "checkOnly" : false,
        "ignoreWarnings" : false,
        "performRetrieve" : false,
        "purgeOnDelete" : false,
        "rollbackOnError" : false,
        "runTests" : null,
        "singlePackage" : true,
        "testLevel" : "RunAllTestsInOrg"
        }
    }
----------------------------BOUNDARY
Content-Disposition: form-data; name="file"; filename="deploy.zip"
Content-Type: application/zip

//Contents of deploy.zip
----------------------------BOUNDARY--
```

Response Body: Deploy Metadata

When an HTTP status code of `201` (Created) is returned, your request has succeeded and resulted in the creation of a deployment that is being processed.

```
{ "id" : "0Afxx00000001VPCAY",
  "deployOptions" :
   { "checkOnly" : false,
     "singlePackage" : false,
     "allowMissingFiles" : false,
     "performRetrieve" : false,
     "autoUpdatePackage" : false,
     "rollbackOnError" : true,
     "ignoreWarnings" : false,
     "purgeOnDelete" : false,
     "runAllTests" : false },
  "deployResult" :
```

```
{ "id" : "0Afxx00000001VPCAY",
  "success" : false,
  "checkOnly" : false,
  "ignoreWarnings" : false,
  "rollbackOnError" : true,
  "status" : "Pending",
  "runTestsEnabled" : false,
  "done" : false } }
```

deployResult Parameters

Parameter	Description
`id`	ID. ID of the component being deployed.
`canceledBy`	ID. The ID of the user who canceled the deployment.
`canceledByName`	String. The full name of the user who canceled the deployment.
`checkOnly`	Boolean. Indicates whether this deployment is used to check the validity of the deployed files without changing the org (`true`) or not (`false`). A check-only deployment doesn't deploy any components or change the organization in any way.
`completedDate`	DateTime. Timestamp for when the deployment process ended.
`createdBy`	ID. The ID of the user who created the deployment.
`createdByName`	String. The full name of the user who created the deployment.
`createdDate`	DateTime. Timestamp for when the deploy request was received.
`details`	DeployDetails. Provides the details of a deployment that is in-progress or ended if `?includeDetails=true` is added as a query to the GET request.
`done`	Boolean. Indicates whether the server finished processing the deploy request for the specified `id`.
`errorMessage`	String. Message corresponding to the values in the `errorStatusCode` field, if any.
`errorStatusCode`	String. If an error occurred during the deploy request, a status code is returned, and the message corresponding to the status code is returned in `errorMessagefield`.
`ignoreWarnings`	Boolean. Optional. Defaults to `false`. Specifies whether a deployment continues even if the deployment generates warnings. Don't set this argument to `true` for deployments to production organizations.
`lastModifiedDate`	DateTime. Timestamp of the last update for the deployment process.
`numberComponentErrors`	Int. The number of components deployed in the deployment process. Use this value with the `numberComponentsTotal` value to get an estimate of the deployment's progress.
`numberComponentsTotal`	Int. The total number of components in the deployment. Use this value with the `numberComponentsDeployed` value to get an estimate of the deployment's progress.

Parameter	Description
`numberTestErrors`	Int. The number of Apex tests that have generated errors during this deployment.
`numberTestsCompleted`	The number of completed Apex tests for this deployment. Use this value with the `numberTestsTotal` value to get an estimate of the deployment's test progress.
`numberTestsTotal`	Int. The total number of Apex tests for this deployment. Use this value with the `numberTestsCompleted` value to get an estimate of the deployment's test progress. The value in this field isn't accurate until the deployment has started running tests for the components being deployed.
`runTestsEnabled`	Boolean. Indicates whether Apex tests were run as part of this deployment (`true`) or not (`false`). Tests are either automatically run as part of a deployment or can be set to run in the deployOptions child object.
`rollbackOnError`	Boolean. Defaults to `true`. Indicates whether any failure causes a complete rollback (`true`) or not (`false`). If `false`, whatever set of actions can be performed without errors are performed, and errors are returned for the remaining actions. This parameter must be set to `true` if you're deploying to a production org.
`startDate`	DateTime. Timestamp for when the deployment process began.
`stateDetail`	String. Indicates which component is being deployed or which Apex test class is running.
`status`	Indicates the current state of the deployment. The valid values are: • `Pending` • `InProgress` • `Succeeded` • `SucceededPartial` • `Failed` • `Canceling` • `Canceled`
`success`	Boolean. Indicates whether the deployment was successful (`true`) or not (`false`).

Check the Status of Your Deployment Using REST Resources

Check the status of your deployment by using passing the deployment request ID in the URL The response body is similar to that returned by the original deployment request, but it includes information about the deployment in progress.

URI

`https://`***`host`***`/services/data/v`***`XX.`***`0/metadata/deployRequest/`***`deployRequestId`***

To include more details in the response, use:

`https://`***`host`***`/services/data/v`***`XX.`***`0/metadata/deployRequest/`***`deployRequestId`***`?includeDetails=true`

Formats

JSON

HTTP Method

GET

Authentication

```
Authorization: Bearer token
```

Response Body: Deploy Metadata

The following example shows the response when `?includeDetails=true` is added as a query to the GET request.

```
    {
        "id" : "0Afxx000000001WCAQ"
        "url" :
"https://host/services/data/vXX.0/metadata/deployRequest/0Afxx000000001WCAQ?includeDetails=true",

        "deployResult" :
            {
            "checkOnly" : "false",
            "ignoreWarnings" : "false",
            "rollbackOnError" : "false",

            "status : "InProgress",
            "numberComponentsDeployed" : "10",
            "numberComponentsTotal" : "1032",
            "numberComponentErrors" : "0",
            "numberTestsCompleted" : "45",
            "numberTestsTotal" : "135",
            "numberTestErrors" : "0",
            "details" :  {
              "componentFailures" : [],
              "componentSuccesses" : [],
                    "retrieveResult" : null,
                    "runTestResults" : {
                    "numRun" : 0,
                    "successes" : [ … ],
                    "failures" : []
                    }
            },

            "createdDate" : "2017-10-10T08:22Z",
            "startDate" : "2017-10-10T08:22Z",
            "lastModifiedDate" : "2017-10-10T08:44Z",
            "completedDate" : "2017-10-10T08:44Z",

            "errorStatusCode" : null,
            "errorMessage" : null,
            "stateDetail" : "Processing Type: Apex Component",

            "createdBy" : "005xx0000001Sv1m",
            "createdByName" : "stephanie stevens",
            "canceledBy" : null,
            "canceledByName" : null,
            "isRunTestsEnabled" : null
            }

       "deployOptions": {
```

```
            "allowMissingFiles" : false,
            "autoUpdatePackage" : false,
            "checkOnly" : true,
            "ignoreWarnings" : false,
            "performRetrieve" : false,
            "purgeOnDelete" : false,
            "rollbackOnError" : false,
            "runTests" : null,
            "singlePackage" : true,
            "testLevel" : "RunAllTestsInOrg"
            }
    }
```

Expect an HTTP status code of `200` (OK) to be returned.

Deploy a Recently Validated Component Set Without Tests

You can deploy components to production in less time by skipping the execution of Apex tests when testing requirements have already been met.

- The components have been validated successfully for the target environment within the last 10 days.
- As part of the validation, Apex tests in the target org have passed.
- Code coverage requirements are met.
  - If all tests in the org or all local tests are run, overall code coverage is at least 75%, and Apex triggers have some coverage.
  - If specific tests are run with the `RunSpecifiedTests` test level, each class and trigger to be deployed is covered by at least 75% individually.

This operation is equivalent to performing a quick deployment of a recent validation on the Deployment Status page in the Salesforce user interface.

To validate but not deploy a set of components when using the `deployRequest` resource, set the `checkOnly` parameter of `deployOptions` to `true`. Note the deployment request ID in the response. Use this ID (associated with a successful validation) later to deploy the component set without repeating the validation.

URI

`https://`***`host`***`/services/data/v`***`XX.`***`0/metadata/deployRequest/`***`validatedDeployRequestId`***

Formats

JSON

HTTP Method

POST

Authentication

`Authorization: Bearer` ***`token`***

Request Body: Deploy a Recently Validated Component Set Without Tests

Note: The HTTP method for deploying a recently validated component set is POST, not PATCH. Using PATCH would create a new deployment.

```
    {
```

```
        "validatedDeployRequestId" : "0Afxx000000001WCAQ"
    }
```

If there is no corresponding deployment package that meets the validation requirements, you receive an HTTP status code of `404` (Not Found). If the validated deployment package is found, the HTTP status code returned is `201` (Created).

Response Body: Deploy a Recently Validated Component Set Without Tests

Note: The response body from the deployment without validation request includes a new request ID, because it is separate from the earlier request for a validation-only deployment.

```
    {
       "validatedDeployRequestId" : "0Afxx000000001WCAQ"
       "id" : "0Afxx000000001WMEM"
      "url" : "https://host/services/data/vXX.0/metadata/deployRequest/0Afxx000000001WMEM",

       "deployOptions" :
             {
             "allowMissingFiles" : false,
             "autoUpdatePackage" : false,
             "checkOnly" : true,
             "ignoreWarnings" : false,
             "performRetrieve" : false,
             "purgeOnDelete" : false,
             "rollbackOnError" : false,
             "runTests" : null,
             "singlePackage" : true,
             "testLevel" : "RunAllTestsInOrg"
}
     }
```

When an HTTP status code of `201` (Created) is returned, your request has succeeded and resulted in the creation of a deployment that is being processed. In the preceding example response body, the ID of the validation-only deployment request is `0Afxx000000001WCAQ`; the ID of the deployment without validation request is `0Afxx000000001WMEM`.

Cancel a Deployment in Progress Using REST

You can request a cancellation of a deployment that's already in progress. Make the cancellation request by patching the status of an ongoing `deployRequest`. The cancellation is processed asynchronously.

URI

`https://host/services/data/vXX.0/metadata/deployRequest/deployRequestId`

Formats

JSON

HTTP Method

PATCH

Authentication

`Authorization: Bearer token`

Request Body: Request Deployment Cancelation

The JSON request body for a deployment cancellation includes a PATCH to the status of the original `deployRequest`.

```
{
 "deployResult":
       {
       "status" : "Canceling"
       }
}
```

Response Body: Request Deployment Cancelation

Because the cancellation request is processed asynchronously, the status shown in the response body can be either `Canceling` or `Canceled`.

```
{
   "id" : "0Afxx000000001WCAQ"
  "url" : "https://host/services/data/vXX.0/metadata/deployRequest/0Afxx000000001WCAQ",

   "deployResult":
         {
         "checkOnly" : "false",
         "ignoreWarnings" : "false",
         "rollbackOnError" : "false",
         "status : "Canceling",  // or Canceled
         "numberComponentsDeployed" : "10",
         "numberComponentsTotal" : "1032",
         "numberComponentErrors" : "0",
         "numberTestsCompleted" : "45",
         "numberTestsTotal" : "135",
         "numberTestErrors" : "0",
         "details" :  {
            "componentFailures" : [],
            "componentSuccesses" : [],
                  "retrieveResult" : null,
                  "runTestResults" : {
                     "numRun" : 0,
                     "successes" : [ … ],
                     "failures" : []
                  }
            },

            "createdDate" : "2017-10-10T08:22Z",
            "startDate" : "2017-10-10T08:22Z",
            "lastModifiedDate" : "2017-10-10T08:44Z",
            "completedDate" : "2017-10-10T08:44Z",
            "errorStatusCode" : null,
            "errorMessage" : null,
            "stateDetail" : "Processing Type: Apex Component",
            "createdBy" : "005xx0000001Sv1m",
            "createdByName" : "steve stevens",
```

```
                "canceledBy" : null,
                "canceledByName" : null,
                "isRunTestsEnabled" : null
                }
        }
```

When an HTTP status code of `202` (Accepted) is returned, your cancelation request is in progress or successful.

Deploy Metadata with REST API in Salesforce CLI

By default, the Salesforce CLI `project deploy start` command uses the Metadata SOAP API to deploy source to your org. You can use the Metadata REST API instead by setting a CLI configuration value or environment variable. Compared with SOAP API, REST API offers faster deployment.

Use the `org-metadata-rest-deploy` Salesforce CLI runtime configuration variable or `SF_ORG_METADATA_REST_DEPLOY` environment variable to set REST API as the default. For more information, see the *Salesforce DX Setup Guide*.

This example uses the configuration value to set the default for your current project:

USER PERMISSIONS

To work with Metadata API from Salesforce CLI:

- Modify Metadata Through Metadata API Functions

  Or

  Modify All Data

```
sf config set org-metadata-rest-deploy true
```

To set the default globally for all your projects, use the `--global` flag:

```
sf config set org-metadata-rest-deploy true --global
```

Note: Only commands that deploy source, such as **project deploy start**, support REST API. Commands that retrieve source, such as **project retrieve start**, always use SOAP API.

Here are the deploy limits. Limits can change without notice.

Feature	Limit
Maximum compressed .zip folder size[1] (SOAP API)	Approximately 39 MB
Maximum uncompressed folder size[2] (SOAP API)	Approximately 600 MB
Maximum number of files in AppExchange packages (REST and SOAP API)	35,000
Maximum number of files in packages (REST and SOAP API)	10,000

[1] Metadata API base-64 encodes components after they're compressed. The resulting .zip file can't exceed 50 MB. Base-64 encoding increases the size of the payload by approximately 22%, so your compressed payload can't exceed approximately 39 MB before encoding.

[2] When using the Ant Migration Tool to deploy an unzipped project, all files in the project are compressed first. The maximum size of uncompressed components in an uncompressed project is 600 MB or less, depending on the files' compression ratio. If the files have a high compression ratio, you can migrate a total of approximately 600 MB because the compressed size would be under 39 MB. However, if the components can't be compressed much, like binary static resources, you can migrate less than 600 MB.

CHAPTER 7 Error Handling

Metadata API calls return error information that your client application can use to identify and resolve runtime errors.

Metadata API provides these types of error handling.

- Since the Metadata API uses the enterprise or partner WSDLs to authenticate, it uses SOAP fault messages defined in those WSDLs for errors resulting from badly formed messages, failed authentication, or similar problems. Each SOAP fault has an associated ExceptionCode. For more details, see Error Handling in the *SOAP API Developer Guide*.
- For errors with the asynchronous create(), update(), and delete() calls, see the error status code in the `statusCode` field in the AsyncResult object for the associated component.
- For errors with the synchronous CRUD calls, see the error status code in the `statusCode` field of the Error object corresponding to each error in the array returned by the `errors` field of the appropriate result object. For example, the result object of createMetadata() is SaveResult.
- For errors with deploy(), see the `problem` and `success` fields in the DeployMessage object for the associated component.
- For errors with retrieve(), see the `problem` field in the RetrieveMessage object for the associated component.

For sample code, see Step 3: Walk Through the Java Sample Code on page 16.

Error Handling for Session Expiration

When you sign on via the `login()` call, a new client session begins and a corresponding unique session ID is generated. Sessions automatically expire after the amount of time specified in the **Security Controls** setup area of the Salesforce application (default two hours). When your session expires, the exception code INVALID_SESSION_ID is returned. If this happens, you must invoke the `login()` call again. For more information about `login()`, see the *SOAP API Developer Guide*.

CHAPTER 8 File-Based Calls

Use file-based calls to deploy or retrieve XML components.

- deploy()
- deployRecentValidation()
- retrieve()

deploy()

Uses file representations of components to create, update, or delete those components in a Salesforce org.

Syntax

```
AsyncResult = metadatabinding.deploy(base64 zipFile, DeployOptions deployOptions)
```

Usage

Use this call to take file representations of components and deploy them into an org by creating, updating, or deleting the components they represent.

Here are the deploy limits. Limits can change without notice.

Feature	Limit
Maximum compressed .zip folder size[1]	Approximately 39 MB
Maximum uncompressed folder size[2]	Approximately 600 MB
Maximum number of files included in a first-generation managed package (1GP). Only 1GP packages that have passed AppExchange Security Review can contain this amount of files.	35,000
Maximum number of files included in an unlocked or second-generation managed package	10,000

[1] Metadata API base-64 encodes components after they're compressed. The resulting .zip file can't exceed 50 MB. Base-64 encoding increases the size of the payload by approximately 22%, so your compressed payload can't exceed approximately 39 MB before encoding.

[2] When using the Ant Migration Tool to deploy an unzipped project, all files in the project are compressed first. The maximum size of uncompressed components in an uncompressed project is 600 MB or less, depending on the files' compression ratio. If the files have a

high compression ratio, you can migrate a total of approximately 600 MB because the compressed size would be under 39 MB. However, if the components can't be compressed much, like binary static resources, you can migrate less than 600 MB.

In API version 29.0, Salesforce improved the deployment status properties and removed the requirement to use `checkStatus()` after a `deploy()` call to get information about deployments. Salesforce continues to support the use of `checkStatus()` when using `deploy()` with API version 28.0 or earlier.

Deploy Components to an Org

The `package.xml` file is a project manifest that lists all the components that you want to retrieve or deploy. You can use `package.xml` to add components. To delete components, add another manifest file. See Deleting Components from an Organization.

For API version 29.0 or later, here's how to deploy (create or update) packaged or unpackaged components.

1. Issue a `deploy()` call to start the asynchronous deployment. An AsyncResult object is returned. Note the value in the `id` field, and use it for the next step.
2. Issue a `checkDeployStatus()` call in a loop until the `done` field of the returned DeployResult contains `true`, which means that the call is completed. The DeployResult object contains information about an in-progress or completed deployment started using the `deploy()` call. When calling `checkDeployStatus()`, pass in the `id` value from the AsyncResult object from the first step.

For API version 28.0 or earlier, here's how to deploy (create or update) packaged or unpackaged components.

1. Issue a `deploy()` call to start the asynchronous deployment. An AsyncResult object is returned. If the call is completed, the `done` field contains `true`. Most often, the call isn't completed quickly enough to be noted in the first result. If it's completed, note the value in the `id` field returned, and skip the next step.
2. If the call isn't complete, issue a `checkStatus()` call in a loop. In the loop, use the value in the `id` field of the AsyncResult object returned by the `deploy()` call in the previous step. Check the AsyncResult object, which is returned until the `done` field contains `true`. The time taken to complete a `deploy()` call depends on the size of the zip file being deployed. Therefore, use a longer wait time between iterations as the size of the zip file increases.
3. Issue a `checkDeployStatus()` call to obtain the results of the `deploy()` call, using the `id` value returned in the first step.

Note: The deployment process locks write-access to resources getting deployed until deployment completes. During deployment, changes made to locked resources or related items can result in errors. Salesforce recommends deployments during off-peak usage time and limiting or postponing changes to your org while deployment is in progress.

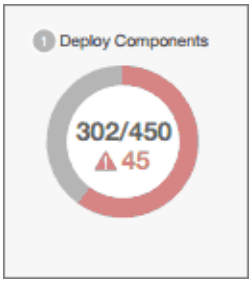
Check the Status of a Deployment

Check the status of a deployment using Metadata API or from Setup. You can check the status of deployments that are in progress or completed in the last 30 days.

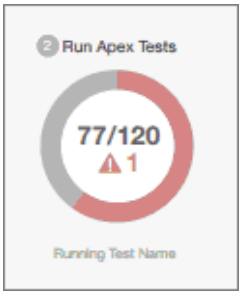
To check the status of a deployment using Metadata API, see checkDeployStatus() on page 68.

To check the status of a deployment from Setup, enter *`Deployment Status`* in the Quick Find box, then select **Deployment Status**.

When running a deployment, the Deployment Status page shows you the real-time progress of your current deployment. This page contains charts that provide a visual representation of the overall deployment progress. The first chart shows how many components have already been deployed out of the total and includes the number of components with errors. For example, the following chart indicates that 302 components were processed successfully out of 450 and there were 45 components with errors.



After all components have been deployed without errors, Apex tests start executing, if necessary or enabled. A second chart shows how many Apex tests have run out of the total number of tests and the number of errors returned. In addition, the chart shows the name of the currently running test. For example, in the following chart, 77 tests have completed execution out of a total of 120, and 1 test failed.



You can initiate multiple deployments, but only one deployment can run at a time. The other deployments remain in the queue waiting to be executed after the current deployment finishes. Queued deployments are listed under Pending Deployments and are not necessarily executed in the order in which they were submitted. To execute deployments in a particular order, submit them one at a time after the previous deployment has completed successfully.

Cancel a Deployment

Cancel a deployment using the Metadata API or from Setup. You can cancel a deployment while it's in progress or in the queue.

To cancel a deployment using Metadata API, see cancelDeploy().

To cancel a deployment from Setup, enter `Deployment Status` in the quick find box, then select **Deployment Status**. Click **Cancel** next to the deployment you want to cancel. The deployment has the status of `Cancel Requested` until the cancellation completes. A canceled deployment is listed in the Failed section.

Permissions

Your client application must be logged in with the Modify Metadata Through Metadata API Functions or Modify All Data permission.

Note: If a user requires access to metadata but not to data, enable the Modify Metadata Through Metadata API Functions permission. Otherwise, enable the Modify All Data permission.

Arguments

Name	Type	Description
`zipFile`	base64	Base 64-encoded binary data. Client applications must encode the binary data as base64.

Name	Type	Description
`deployOptions`	DeployOptions	Encapsulates options for determining which packages or files are deployed.

DeployOptions

The following deployment options can be selected for this call:

Name	Type	Description
`allowMissingFiles`	boolean	If files that are specified in `package.xml` aren't in the `.zip` file, specifies whether a deployment can still succeed. Don't set this argument for deployment to *production orgs*.
`autoUpdatePackage`	boolean	If a file is in the `.zip` file but not specified in `package.xml`, specifies whether the file is automatically added to the package. A `retrieve()` is issued with the updated `package.xml` that includes the `.zip` file. Don't set this argument for deployment to *production orgs*.
`checkOnly`	boolean	Defaults to `false`. Set to `true` to perform a test deployment (validation) of components without saving the components in the target org. A validation enables you to verify the results of tests that would be generated in a deployment, but doesn't commit any changes. After a validation finishes with passing tests, sometimes it can qualify for deployment without rerunning tests. See deployRecentValidation(). If you change a field type from Master-Detail to Lookup or vice versa, the change isn't supported when using the `checkOnly` option to test a deployment. This change isn't supported for test deployments to avoid permanently altering your data. If a change that isn't supported for test deployments is included in a deployment package, the test deployment fails and issues an error. If your deployment package changes a field type from Master-Detail to Lookup or vice versa, you can still validate the changes before you deploy to production. Perform a full deployment to another test sandbox. A full deployment includes a validation of the changes as part of the deployment process. A Metadata API deployment that includes Master-Detail relationships deletes all detail records in the Recycle Bin in the following cases. 1. For a deployment with a new Master-Detail field, soft delete (send to the Recycle Bin) all detail records before proceeding to deploy the Master-Detail field, or the deployment fails. During the deployment, detail records

Name	Type	Description
		are permanently deleted from the Recycle Bin and can't be recovered. 2. For a deployment that converts a Lookup field relationship to a Master-Detail relationship, detail records must reference a master record or be soft-deleted (sent to the Recycle Bin) for the deployment to succeed. However, a successful deployment permanently deletes any detail records in the Recycle Bin.
`ignoreWarnings`	boolean	Indicates whether deployments with warnings complete successfully (`true`) or not (`false`). Defaults to `false`. The DeployMessage object for a warning contains the following values: • `problemType`—`Warning` • `problem`—The text of the warning If a warning occurs and `ignoreWarnings` is set to `true`, the `success` field in DeployMessage is `true`. If `ignoreWarnings` is set to `false`, `success` is set to `false` and the warning is treated like an error. This field is available in API version 18.0 and later. Before version 18.0, there was no distinction between warnings and errors. All problems were treated as errors and prevented a successful deployment.
`performRetrieve`	boolean	Indicates whether a `retrieve()` call is performed immediately after the deployment (`true`) or not (`false`). Set to `true` to retrieve whatever was deployed.
`purgeOnDelete`	boolean	If `true`, the deleted components in the `destructiveChanges.xml` manifest file aren't stored in the Recycle Bin. Instead, they become immediately eligible for deletion. This field is available in API version 22.0 and later. This option only works in Developer Edition or sandbox orgs. It doesn't work in production orgs. When you delete a roll-up summary field using Metadata API, the field isn't saved in the Recycle Bin. The field is purged even if you don't set the `purgeOnDelete` deployment option to `true`.
`rollbackOnError`	boolean	Indicates whether any failure causes a complete rollback (`true`) or not (`false`). If `false`, whatever actions can be performed without errors are performed, and errors are returned for the remaining actions. This parameter must be

Name	Type	Description
		set to `true` if you're deploying to a production org. The default is `false`.
`runAllTests`	boolean	(Deprecated and only available in API version 33.0 and earlier.) This field defaults to `false`. Set to `true` to run all Apex tests after deployment, including tests that originate from installed managed packages. Apex tests that run as part of a deployment always run synchronously and serially.
`runTests`	string[]	A list of Apex tests to run during deployment. Specify the class name, one name per instance. The class name can also specify a namespace with a dot notation. For more information, see Running a Subset of Tests in a Deployment. To use this option, set `testLevel` to `RunSpecifiedTests`.
`singlePackage`	boolean	Indicates whether the specified `.zip` file points to a directory structure with a single package (`true`) or a set of packages (`false`).
`testLevel`	TestLevel (enumeration of type string)	Optional. Specifies which tests are run as part of a deployment. The test level is enforced regardless of the types of components that are present in the deployment package. Valid values are: • `NoTestRun`—No tests are run. This test level applies only to deployments to development environments, such as sandbox, Developer Edition, or trial organizations. This test level is the default for development environments. • `RunSpecifiedTests`—Only the tests that you specify in the `runTests` option are run. Code coverage requirements differ from the default coverage requirements when using this test level. Each class and trigger in the deployment package must be covered by the executed tests for a minimum of 75% code coverage. This coverage is computed for each class and triggers individually and is different than the overall coverage percentage. • `RunLocalTests`—All tests in your org are run, except the ones that originate from installed managed and unlocked packages. This test level is the default for production deployments that include Apex classes or triggers. • `RunAllTestsInOrg`—All tests are run. The tests include all tests in your org, including tests of managed packages. If you don't specify a test level, the default test execution behavior is used. See Running Tests in a Deployment.

Name	Type	Description
		Apex tests that run as part of a deployment always run synchronously and serially. This field is available in API version 34.0 and later.

Response

AsyncResult

Sample Code—Java

This sample shows how to deploy components in a zip file. See the retrieve() sample code for details on how to retrieve a zip file.

```
package com.doc.samples;

import java.io.*;

import java.rmi.RemoteException;

import com.sforce.soap.metadata.AsyncResult;
import com.sforce.soap.metadata.DeployDetails;
import com.sforce.soap.metadata.MetadataConnection;
import com.sforce.soap.metadata.DeployOptions;
import com.sforce.soap.metadata.DeployResult;
import com.sforce.soap.metadata.DeployMessage;
import com.sforce.soap.metadata.RunTestsResult;
import com.sforce.soap.metadata.RunTestFailure;
import com.sforce.soap.metadata.CodeCoverageWarning;
import com.sforce.soap.enterprise.LoginResult;
import com.sforce.soap.enterprise.EnterpriseConnection;
import com.sforce.ws.ConnectionException;
import com.sforce.ws.ConnectorConfig;

/**
 * Deploy a zip file of metadata components.
 * Prerequisite: Have a deploy.zip file that includes a package.xml manifest file that
 * details the contents of the zip file.
 */
public class DeploySample {
    // binding for the metadata WSDL used for making metadata API calls
    private MetadataConnection metadataConnection;

    static BufferedReader rdr = new BufferedReader(new InputStreamReader(System.in));

    private static final String ZIP_FILE = "deploy.zip";

    // one second in milliseconds
    private static final long ONE_SECOND = 1000;
    // maximum number of attempts to deploy the zip file
    private static final int MAX_NUM_POLL_REQUESTS = 50;
```

```
public static void main(String[] args) throws Exception {
    final String USERNAME = "user@company.com";
    // This is only a sample. Hard coding passwords in source files is a bad practice.

    final String PASSWORD = "password";
    final String URL = "https://login.salesforce.com/services/Soap/c/29.0";

    DeploySample sample = new DeploySample(USERNAME, PASSWORD, URL);
    sample.deployZip();
}

public DeploySample(String username, String password, String loginUrl)
        throws ConnectionException {
    createMetadataConnection(username, password, loginUrl);
}

public void deployZip()
    throws RemoteException, Exception
{
    byte zipBytes[] = readZipFile();
    DeployOptions deployOptions = new DeployOptions();
    deployOptions.setPerformRetrieve(false);
    deployOptions.setRollbackOnError(true);
    AsyncResult asyncResult = metadataConnection.deploy(zipBytes, deployOptions);
    String asyncResultId = asyncResult.getId();

    // Wait for the deploy to complete
    int poll = 0;
    long waitTimeMilliSecs = ONE_SECOND;
    DeployResult deployResult = null;
    boolean fetchDetails;
    do {
        Thread.sleep(waitTimeMilliSecs);
        // double the wait time for the next iteration
        waitTimeMilliSecs *= 2;
        if (poll++ > MAX_NUM_POLL_REQUESTS) {
            throw new Exception("Request timed out. If this is a large set " +
                    "of metadata components, check that the time allowed by " +
                    "MAX_NUM_POLL_REQUESTS is sufficient.");
        }

        // Fetch in-progress details once for every 3 polls
        fetchDetails = (poll % 3 == 0);
      deployResult = metadataConnection.checkDeployStatus(asyncResultId, fetchDetails);

        System.out.println("Status is: " + deployResult.getStatus());
        if (!deployResult.isDone() && fetchDetails) {
            printErrors(deployResult, "Failures for deployment in progress:\n");
        }
    }
    while (!deployResult.isDone());

    if (!deployResult.isSuccess() && deployResult.getErrorStatusCode() != null) {
```

```
            throw new Exception(deployResult.getErrorStatusCode() + " msg: " +
                    deployResult.getErrorMessage());
        }

        if (!fetchDetails) {
            // Get the final result with details if we didn't do it in the last attempt.
            deployResult = metadataConnection.checkDeployStatus(asyncResultId, true);
        }

        if (!deployResult.isSuccess()) {
            printErrors(deployResult, "Final list of failures:\n");
            throw new Exception("The files were not successfully deployed");
        }

        System.out.println("The file " + ZIP_FILE + " was successfully deployed");
    }

    /**
     * Read the zip file contents into a byte array.
     * @return byte[]
     * @throws Exception - if cannot find the zip file to deploy
     */
    private byte[] readZipFile()
        throws Exception
    {
        // We assume here that you have a deploy.zip file.
        // See the retrieve sample for how to retrieve a zip file.
        File deployZip = new File(ZIP_FILE);
        if (!deployZip.exists() || !deployZip.isFile())
            throw new Exception("Cannot find the zip file to deploy. Looking for " +
                    deployZip.getAbsolutePath());

        FileInputStream fos = new FileInputStream(deployZip);
        ByteArrayOutputStream bos = new ByteArrayOutputStream();
        int readbyte = -1;
        while ((readbyte = fos.read()) != -1)  {
            bos.write(readbyte);
        }
        fos.close();
        bos.close();
        return bos.toByteArray();
    }


    /**
     * Print out any errors, if any, related to the deploy.
     * @param result - DeployResult
     */
    private void printErrors(DeployResult result, String messageHeader)
    {
        DeployDetails deployDetails = result.getDetails();

        StringBuilder errorMessageBuilder = new StringBuilder();
        if (deployDetails != null) {
```

```
            DeployMessage[] componentFailures = deployDetails.getComponentFailures();
            for (DeployMessage message : componentFailures) {
                String loc = (message.getLineNumber() == 0 ? "" :
                    ("(" + message.getLineNumber() + "," +
                            message.getColumnNumber() + ")"));
                if (loc.length() == 0
                        && !message.getFileName().equals(message.getFullName())) {
                    loc = "(" + message.getFullName() + ")";
                }
                errorMessageBuilder.append(message.getFileName() + loc + ":" +
                        message.getProblem()).append('\n');
            }
            RunTestsResult rtr = deployDetails.getRunTestResult();
            if (rtr.getFailures() != null) {
                for (RunTestFailure failure : rtr.getFailures()) {
                    String n = (failure.getNamespace() == null ? "" :
                        (failure.getNamespace() + ".")) + failure.getName();
                    errorMessageBuilder.append("Test failure, method: " + n + "." +
                            failure.getMethodName() + " -- " +
                            failure.getMessage() + " stack " +
                            failure.getStackTrace() + "\n\n");
                }
            }
            if (rtr.getCodeCoverageWarnings() != null) {
                for (CodeCoverageWarning ccw : rtr.getCodeCoverageWarnings()) {
                    errorMessageBuilder.append("Code coverage issue");
                    if (ccw.getName() != null) {
                        String n = (ccw.getNamespace() == null ? "" :
                            (ccw.getNamespace() + ".")) + ccw.getName();
                        errorMessageBuilder.append(", class: " + n);
                    }
                    errorMessageBuilder.append(" -- " + ccw.getMessage() + "\n");
                }
            }
        }

        if (errorMessageBuilder.length() > 0) {
            errorMessageBuilder.insert(0, messageHeader);
            System.out.println(errorMessageBuilder.toString());
        }
    }

    private void createMetadataConnection(
        final String username,
        final String password,
        final String loginUrl) throws ConnectionException {

        final ConnectorConfig loginConfig = new ConnectorConfig();
        loginConfig.setAuthEndpoint(loginUrl);
        loginConfig.setServiceEndpoint(loginUrl);
        loginConfig.setManualLogin(true);
        LoginResult loginResult = (new EnterpriseConnection(loginConfig)).login(
                username, password);
```

```
        final ConnectorConfig metadataConfig = new ConnectorConfig();
        metadataConfig.setServiceEndpoint(loginResult.getMetadataServerUrl());
        metadataConfig.setSessionId(loginResult.getSessionId());
        this.metadataConnection = new MetadataConnection(metadataConfig);
    }

}
```

1. Deleting Components from an Organization
   To delete components, perform a deployment with the `deploy()` call by using a destructive changes manifest file that lists the components to remove from your organization. You can perform a deployment that only deletes components, or a deployment that deletes and adds components. In API version 33.0 and later, you can specify components to delete before and after other components are added or updated. In earlier API versions, if deletions and additions are specified for the same deployment, the `deploy()` call performs the deletions first.
2. checkDeployStatus()
   Checks the status of declarative metadata call `deploy()`.
3. cancelDeploy()
   Cancels a deployment that hasn't completed yet.

SEE ALSO:

Running Tests in a Deployment

Deleting Components from an Organization

To delete components, perform a deployment with the `deploy()` call by using a destructive changes manifest file that lists the components to remove from your organization. You can perform a deployment that only deletes components, or a deployment that deletes and adds components. In API version 33.0 and later, you can specify components to delete before and after other components are added or updated. In earlier API versions, if deletions and additions are specified for the same deployment, the `deploy()` call performs the deletions first.

Deleting Components in a Deployment

To delete components, use the same procedure as with deploying components, but also include a delete manifest file that's named `destructiveChanges.xml` and list the components to delete in this manifest. The format of this manifest is the same as `package.xml` except that wildcards aren't supported.

Note: You can't use `destructiveChanges.xml` to delete items that are associated with an active Lightning page, such as a custom object, a component on the page, or the page itself. First, you must remove the page's action override by deactivating it in the Lightning App Builder.

The following sample `destructiveChanges.xml` file names a single custom object to be deleted:

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>MyCustomObject__c</members>
        <name>CustomObject</name>
    </types>
</Package>
```

To deploy the destructive changes, you must also have a `package.xml` file that lists no components to deploy, includes the API version, and is in the same directory as `destructiveChanges.xml`:

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <version>64.0</version>
</Package>
```

Note:

- To bypass the Recycle Bin, set the `purgeOnDelete` option to `true`.
- When you delete a roll-up summary field using Metadata API, the field isn't saved in the Recycle Bin. The field is purged even if you don't set the `purgeOnDelete` deployment option to `true`.
- If you try to delete some components that don't exist in the organization, the rest of the deletions are still attempted.

Adding and Deleting Components in a Single Deployment

You can perform a deployment that specifies components to delete in `destructiveChanges.xml` and components to add or update in `package.xml`. The process is the same as with performing a delete-only deployment except that `package.xml` contains the components to add or update.

By default, deletions are processed before component additions. In API version 33.0 and later, you can specify components to be deleted before and after component additions. The process is the same as with performing a delete-only deployment except that the name of the deletion manifest file is different.

- To delete components *before* adding or updating other components, create a manifest file that's named `destructiveChangesPre.xml` and include the components to delete.
- To delete components *after* adding or updating other components, create a manifest file that's named `destructiveChangesPost.xml` and include the components to delete.

The ability to specify when deletions are processed is useful when you're deleting components with dependencies. For example, if a custom object is referenced in an Apex class, you can't delete it unless you modify the Apex class first to remove the dependency on the custom object. In this example, you can perform a single deployment that updates the Apex class to clear the dependency and then deletes the custom object by using `destructiveChangesPost.xml`. The following are samples of the `package.xml` and `destructiveChangesPost.xml` manifests that would be used in this example.

Sample `package.xml`, which specifies the class to update:

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>SampleClass</members>
        <name>ApexClass</name>
    </types>
    <version>64.0</version>
</Package>
```

Sample `destructiveChangesPost.xml`, which specifies the custom object to delete after the class update:

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>MyCustomObject__c</members>
        <name>CustomObject</name>
```

```
    </types>
</Package>
```

Note: The API version that the deployment uses is the API version that's specified in `package.xml`.

checkDeployStatus()

Checks the status of declarative metadata call `deploy()`.

Syntax

```
DeployResult = metadatabinding.checkDeployStatus(ID id, includeDetails boolean);
```

Usage

`checkDeployStatus` is used as part of the process for deploying packaged or unpackaged components to an organization:

1. Issue a `deploy()` call to start the asynchronous deployment. An AsyncResult object is returned. Note the value in the `id` field, and use it for the next step.
2. Issue a `checkDeployStatus()` call in a loop until the `done` field of the returned DeployResult contains `true`, which means that the call is completed. The DeployResult object contains information about an in-progress or completed deployment started using the `deploy()` call. When calling `checkDeployStatus()`, pass in the `id` value from the AsyncResult object from the first step.

Note: Calls to `checkDeployStatus()` don't count against the API request limits and allocations.

In API version 29.0, Salesforce improved the deployment status properties and removed the requirement to use `checkStatus()` after a `deploy()` call to get information about deployments. Salesforce continues to support the use of `checkStatus()` when using `deploy()` with API version 28.0 or earlier.

Sample Code—Java

See the `deploy()` sample code for sample usage of this call.

Arguments

Name	Type	Description
`id`	ID	ID obtained from an AsyncResult object returned by deploy() or a subsequent checkStatus() call.
`includeDetails`	boolean	Sets the DeployResult object to include DeployDetails information (`true`) or not (`false`). The default is `false`. Available in API version 29.0 and later.

Response

DeployResult

cancelDeploy()

Cancels a deployment that hasn't completed yet.

Syntax

```
CancelDeployResult = metadatabinding.cancelDeploy(string id)
```

Usage

Use the `cancelDeploy()` operation to cancel a deployment in your organization started by the deploy() operation, which includes deployments started by the Lightning Platform Migration Tool and the Lightning Platform IDE. The deployment can be in a queue waiting to get started, or can be in progress. This operation takes the ID of the deployment you want to cancel and returns a CancelDeployResult object. When the deployment is in the queue and hasn't started yet, calling `cancelDeploy()` cancels the deployment immediately. When the deployment has started and is in progress, sometimes it doesn't get canceled immediately, so call checkDeployStatus() to check the status of the cancellation.

Cancel a deployment using these steps.

1. Obtain the ID of the deployment you want to cancel. For example, you can obtain the ID from the deploy() call in the `AsyncResult` object `id` field. Alternatively, you can obtain the ID in the Salesforce user interface from Setup by entering *`Deployment Status`* in the `Quick Find` box, selecting **Deployment Status**, and then noting the ID of a deployment started by the API.
2. Issue a `cancelDeploy()` call to start the cancellation process. This call returns a `CancelDeployResult` object.
3. Check the value in the `done` field of the returned `CancelDeployResult`. If the `done` field value is `true`, the deployment has been canceled and you're done. If the `done` field value is `false`, the cancellation is in progress. To check the cancellation status, follow these steps.
   a. Call checkDeployStatus() using the deployment ID you obtained earlier.
   b. In the returned DeployResult object, check the `status` field. If the status is `Canceling`, the cancellation is still in progress, and repeat steps a and b. Otherwise, if the status is `Canceled`, the deployment has been canceled and you're done.

The `deploy()` operation throws these API faults.

INVALID_ID_FIELD with the message `Invalid deploy ID`
The specified ID argument doesn't correspond to a valid deployment.

INVALID_ID_FIELD with the message `Deployment already completed`
The specified deployment has already completed.

Version

Available in API version 30.0 and later.

Permissions

Your client application must be logged in with the Modify Metadata Through Metadata API Functions or Modify All Data permission.

Note: If a user requires access to metadata but not to data, enable the Modify Metadata Through Metadata API Functions on page 7 permission. Otherwise, enable the Modify All Data permission.

Arguments

Name	Type	Description
id	string	The ID of the deployment to cancel.

Response

CancelDeployResult

Sample Code—Java

This sample shows how to cancel a deployment. The sample calls `cancelDeploy()` by passing it a given deployment ID. Next, it checks whether the cancellation has completed, and if not, calls `checkDeployStatus` in a loop.

```
public void cancelDeploy(String asyncId) throws Exception {
    // Issue the deployment cancellation request
    CancelDeployResult result = metadataConnection.cancelDeploy(asyncId);

    // If the deployment cancellation completed, write a message to the output.
    if (result.isDone()) {
        System.out.println("Your deployment was canceled successfully!");
    }
    else {
        // The deployment cancellation is still in progress, so get a new status
        DeployResult deployResult = metadataConnection.checkDeployStatus(asyncId, false);


        // Check whether the deployment is done. If not done, this means
        // that the cancellation is still in progress and the status is Canceling.

        while (!deployResult.isDone()) {
            // Assert that the deployment status is Canceling
            assert deployResult.getStatus() == DeployStatus.Canceling;
            // Wait 2 seconds
            Thread.sleep(2000);
            // Get the deployment status again
            deployResult = metadataConnection.checkDeployStatus(asyncId, false);
        }

        // The deployment is done. Write the status to the output.
        // (When the deployment is done, the cancellation should have completed
        // and the status should be Canceled. However, in very rare cases,
        // the deployment can complete before it is canceled.)
        System.out.println("Final deploy status = >" + deployResult.getStatus());
    }
}
```

deployRecentValidation()

Deploys a recently validated component set without running Apex tests.

Syntax

```
string = metadatabinding.deployRecentValidation(ID validationID)
```

Usage

Use `deployRecentValidation()` to deploy your components to production in less time by skipping the execution of Apex tests. Ensure that the following requirements are met before deploying a recent validation.

- The components have been validated successfully for the target environment within the last 10 days.
- As part of the validation, Apex tests in the target org have passed.
- Code coverage requirements are met.
  - If all tests in the org or all local tests are run, overall code coverage is at least 75%, and Apex triggers have some coverage.
  - If specific tests are run with the `RunSpecifiedTests` test level, each class and trigger that was deployed is covered by at least 75% individually.

This call is equivalent to performing a quick deployment of a recent validation on the Deployment Status page in the Salesforce user interface.

Before you call `deployRecentValidation()`, your organization must have a validation that was recently run. You can run a validation on a set of components by calling deploy() with the `checkOnly` property of the *`deployOptions`* parameter set to `true`. Note the ID that you obtained from the `deploy()` call. You'll use this ID for the `deployRecentValidation()` call in the next step.

After you've run a validation successfully, use these steps to quick-deploy the validation to the same target environment.

1. To start an asynchronous quick deployment, call `deployRecentValidation()` and pass it the ID of a recent validation. This ID is obtained from the previous deploy() call. The `deployRecentValidation()` call returns the ID of the quick deployment. Note this value. You'll use it in the next step.
2. Check for the completion of the call. This process is similar to that of deploy(). Issue a checkDeployStatus() call in a loop until the `done` field of the returned DeployResult contains `true`, which means that the call is completed. The DeployResult object contains information about an in-progress or completed deployment that was started by using the `deployRecentValidation()` call. When calling checkDeployStatus(), pass in the ID value that you obtained in the first step.

Version

Available in API version 33.0 and later.

Arguments

Name	Type	Description
`validationID`	string	The ID of a recent validation.

Response

Type: string

The ID of the quick deployment.

Sample Code—Java

```
package com.salesforce.test.metadata;

import java.rmi.RemoteException;

import com.sforce.soap.metadata.CodeCoverageWarning;
import com.sforce.soap.metadata.DeployDetails;
import com.sforce.soap.metadata.DeployMessage;
import com.sforce.soap.metadata.DeployResult;
import com.sforce.soap.metadata.MetadataConnection;
import com.sforce.soap.metadata.RunTestFailure;
import com.sforce.soap.metadata.RunTestsResult;
import com.sforce.soap.partner.Connector;
import com.sforce.ws.ConnectionException;
import com.sforce.ws.ConnectorConfig;

/**
 * Quick-deploy a recent validation.
 * Prerequisite: A successful validation (check-only deploy) has been done in the org
recently.
 */
public class DeployRecentValidationSample {
    // binding for the metadata WSDL used for making metadata API calls
    private MetadataConnection metadataConnection;

    // one second in milliseconds
    private static final long ONE_SECOND = 1000;
    // maximum number of attempts to deploy the zip file
    private static final int MAX_NUM_POLL_REQUESTS = 50;

    public static void main(String[] args) throws Exception {
        final String USERNAME = args[0];
        final String PASSWORD = args[1];
        final String URL = args[2];

        final String recentValidationId = args[3];

        DeployRecentValidationSample sample = new DeployRecentValidationSample(
                            USERNAME, PASSWORD, URL);
        sample.deployRecentValidation(recentValidationId);
    }

   public DeployRecentValidationSample(String username, String password, String loginUrl)

            throws ConnectionException {
        createMetadataConnection(username, password, loginUrl);
    }

    public void deployRecentValidation(String recentValidationId)
        throws RemoteException, Exception
    {
       String asyncResultId = metadataConnection.deployRecentValidation(recentValidationId);
```

```
    // Wait for the deploy to complete
    int poll = 0;
    long waitTimeMilliSecs = ONE_SECOND;
    DeployResult deployResult = null;
    boolean fetchDetails;
    do {
        Thread.sleep(waitTimeMilliSecs);
        // double the wait time for the next iteration
        waitTimeMilliSecs *= 2;
        if (poll++ > MAX_NUM_POLL_REQUESTS) {
            throw new Exception("Request timed out. If this is a large set " +
                    "of metadata components, check that the time allowed by " +
                    "MAX_NUM_POLL_REQUESTS is sufficient.");
        }

        // Fetch in-progress details once for every 3 polls
        fetchDetails = (poll % 3 == 0);
      deployResult = metadataConnection.checkDeployStatus(asyncResultId, fetchDetails);

        System.out.println("Status is: " + deployResult.getStatus());
        if (!deployResult.isDone() && fetchDetails) {
            printErrors(deployResult, "Failures for deployment in progress:\n");
        }
    }
    while (!deployResult.isDone());

    if (!deployResult.isSuccess() && deployResult.getErrorStatusCode() != null) {
        throw new Exception(deployResult.getErrorStatusCode() + " msg: " +
                deployResult.getErrorMessage());
    }

    if (!fetchDetails) {
        // Get the final result with details if we didn't do it in the last attempt.
        deployResult = metadataConnection.checkDeployStatus(asyncResultId, true);
    }

    if (!deployResult.isSuccess()) {
        printErrors(deployResult, "Final list of failures:\n");
        throw new Exception("The files were not successfully deployed");
    }

    System.out.println("The recent validation " + recentValidationId +
                       " was successfully deployed");
}

/**
 * Print out any errors, if any, related to the deploy.
 * @param result - DeployResult
 */
private void printErrors(DeployResult result, String messageHeader)
{
    DeployDetails deployDetails = result.getDetails();
```

```
        StringBuilder errorMessageBuilder = new StringBuilder();
        if (deployDetails != null) {
            DeployMessage[] componentFailures = deployDetails.getComponentFailures();
            for (DeployMessage message : componentFailures) {
                String loc = (message.getLineNumber() == 0 ? "" :
                    ("(" + message.getLineNumber() + "," +
                            message.getColumnNumber() + ")"));
                if (loc.length() == 0
                        && !message.getFileName().equals(message.getFullName())) {
                    loc = "(" + message.getFullName() + ")";
                }
                errorMessageBuilder.append(message.getFileName() + loc + ":" +
                        message.getProblem()).append('\n');
            }
            RunTestsResult rtr = deployDetails.getRunTestResult();
            if (rtr.getFailures() != null) {
                for (RunTestFailure failure : rtr.getFailures()) {
                    String n = (failure.getNamespace() == null ? "" :
                        (failure.getNamespace() + ".")) + failure.getName();
                    errorMessageBuilder.append("Test failure, method: " + n + "." +
                            failure.getMethodName() + " -- " +
                            failure.getMessage() + " stack " +
                            failure.getStackTrace() + "\n\n");
                }
            }
            if (rtr.getCodeCoverageWarnings() != null) {
                for (CodeCoverageWarning ccw : rtr.getCodeCoverageWarnings()) {
                    errorMessageBuilder.append("Code coverage issue");
                    if (ccw.getName() != null) {
                        String n = (ccw.getNamespace() == null ? "" :
                            (ccw.getNamespace() + ".")) + ccw.getName();
                        errorMessageBuilder.append(", class: " + n);
                    }
                    errorMessageBuilder.append(" -- " + ccw.getMessage() + "\n");
                }
            }
        }

        if (errorMessageBuilder.length() > 0) {
            errorMessageBuilder.insert(0, messageHeader);
            System.out.println(errorMessageBuilder.toString());
        }
    }

    private void createMetadataConnection(
        final String username,
        final String password,
        final String loginUrl) throws ConnectionException {

        final ConnectorConfig loginConfig = new ConnectorConfig();
        loginConfig.setUsername(username);
        loginConfig.setPassword(password);
        loginConfig.setAuthEndpoint(loginUrl);
```

```
        Connector.newConnection(loginConfig);


        final ConnectorConfig metadataConfig = new ConnectorConfig();
        metadataConfig.setServiceEndpoint(
            loginConfig.getServiceEndpoint().replace("/u/", "/m/"));
        metadataConfig.setSessionId(loginConfig.getSessionId());
        this.metadataConnection = com.sforce.soap.metadata.Connector.
            newConnection(metadataConfig);
    }

}
```

`retrieve()`

The `retrieve()` call retrieves XML file representations of components in an organization.

Syntax

```
AsyncResult = metadatabinding.retrieve(RetrieveRequest)
```

Usage

Use this call to retrieve file representations of components in an organization.

Here are the deploy limits. Limits can change without notice.

Feature	Limit
Maximum compressed .zip folder size[1]	Approximately 39 MB
Maximum uncompressed folder size[2]	Approximately 600 MB
Maximum number of files in AppExchange packages	35,000
Maximum number of files in packages	10,000

Note: You can perform a **retrieve()** call for a big object only if its index is defined. If a big object is created in Setup and doesn't yet have an index defined, you can't retrieve it.

[1] Metadata API base-64 encodes components after they're compressed. The resulting .zip file can't exceed 50 MB. Base-64 encoding increases the size of the payload by approximately 22%, so your compressed payload can't exceed approximately 39 MB before encoding.

[2] When using the Ant Migration Tool to deploy an unzipped project, all files in the project are compressed first. The maximum size of uncompressed components in an uncompressed project is 600 MB or less, depending on the files' compression ratio. If the files have a high compression ratio, you can migrate a total of approximately 600 MB because the compressed size would be under 39 MB. However, if the components can't be compressed much, like binary static resources, you can migrate less than 600 MB.

In API version 31.0 and later, the process of making a `retrieve()` call has been simplified. You no longer have to call checkStatus() after a `retrieve()` call to obtain the status of the retrieve operation. Instead, make calls to checkRetrieveStatus() only. If the retrieve

operation is in progress, call checkRetrieveStatus() again until the retrieve operation is completed. The checkStatus() call is still supported in versions API version 30.0 or earlier, but isn't available in API version 31.0 and later.

For API version 31.0 or later, retrieve packaged or unpackaged components by using the following steps.

1. Issue a `retrieve()` call to start the asynchronous retrieval. An AsyncResult object is returned. Note the value in the `id` field, and use it for the next step.
2. Issue a checkRetrieveStatus() call, and pass in the `id` value from the AsyncResult object from the first step. Check the value of the `done` field of the returned RetrieveResult. If it's `true`, the call is completed, and you proceed to the next step. Otherwise, repeat this step to call checkRetrieveStatus() again until the `done` field is `true`.
3. Retrieve the zip file (`zipFile`) field and other desired fields from RetrieveResult, which the final call to checkRetrieveStatus() returned in the previous step.

For API version 30.0 or earlier, retrieve packaged or unpackaged components by using the following steps.

1. Issue a `retrieve()` call to start the asynchronous retrieval. An AsyncResult object is returned. If the call is completed, the `done` field contains `true`. Most often, the call isn't completed quickly enough to be noted in the result. If it's completed, note the value in the `id` field returned, and skip the next step.
2. If the call isn't completed, issue a checkStatus() call in a loop using the value in the `id` field of the AsyncResult object, returned by the `retrieve()` call in the previous step. Check the AsyncResult object returned until the `done` field contains `true`. The time taken to complete a `retrieve()` call depends on the size of the zip file being deployed, so use a longer wait time between iterations as the size of the zip file increases.
3. Issue a checkRetrieveStatus() call to obtain the results of the `retrieve()` call, using the `id` value returned in the first step.

For examples of manifest files, see Sample package.xml Manifest Files.

Permissions

Your client application must be logged in with the Modify Metadata Through Metadata API Functions or Modify All Data permission.

Note: If a user requires access to metadata but not to data, enable the Modify Metadata Through Metadata API Functions on page 7 permission. Otherwise, enable the Modify All Data permission.

Arguments

Name	Type	Description
`retrieveRequest`	RetrieveRequest	Encapsulates options for determining which packages or files are retrieved.

Response

AsyncResult

Sample Code—Java

This sample shows how to retrieve components into a zip file. See the `deploy()` sample code for details on how to deploy a zip file.

Note: This sample requires API version 34.0 or later.

```
package com.doc.samples;

import java.io.*;
import java.util.*;
import java.nio.ByteBuffer;
import java.nio.channels.*;
import java.rmi.RemoteException;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import org.xml.sax.SAXException;

import com.sforce.soap.metadata.AsyncResult;
import com.sforce.soap.metadata.MetadataConnection;
import com.sforce.soap.enterprise.EnterpriseConnection;
import com.sforce.soap.metadata.RetrieveMessage;
import com.sforce.soap.metadata.RetrieveRequest;
import com.sforce.soap.metadata.RetrieveResult;
import com.sforce.soap.metadata.RetrieveStatus;
import com.sforce.soap.enterprise.LoginResult;
import com.sforce.ws.ConnectionException;
import com.sforce.ws.ConnectorConfig;
import com.sforce.soap.metadata.PackageTypeMembers;

public class RetrieveSample {

    // Binding for the metadata WSDL used for making metadata API calls
    private MetadataConnection metadataConnection;

    static BufferedReader rdr = new BufferedReader(new InputStreamReader(System.in));

    // one second in milliseconds
    private static final long ONE_SECOND = 1000;
    // maximum number of attempts to retrieve the results
    private static final int MAX_NUM_POLL_REQUESTS = 50;

    // manifest file that controls which components get retrieved
    private static final String MANIFEST_FILE = "package.xml";

    private static final double API_VERSION = 31.0;

    public static void main(String[] args) throws Exception {
        final String USERNAME = "user@company.com";
        // This is only a sample. Hard coding passwords in source files is a bad practice.

        final String PASSWORD = "password";
        final String URL = "https://login.salesforce.com/services/Soap/c/31.0";

        RetrieveSample sample = new RetrieveSample(USERNAME, PASSWORD, URL);
```

```
        sample.retrieveZip();
    }

    public RetrieveSample(String username, String password, String loginUrl)
            throws ConnectionException {
        createMetadataConnection(username, password, loginUrl);
    }


    private void retrieveZip() throws RemoteException, Exception
    {
        RetrieveRequest retrieveRequest = new RetrieveRequest();
        // The version in package.xml overrides the version in RetrieveRequest
        retrieveRequest.setApiVersion(API_VERSION);
        setUnpackaged(retrieveRequest);

        // Start the retrieve operation
        AsyncResult asyncResult = metadataConnection.retrieve(retrieveRequest);
        String asyncResultId = asyncResult.getId();

        // Wait for the retrieve to complete
        int poll = 0;
        long waitTimeMilliSecs = ONE_SECOND;
        RetrieveResult result = null;
        do {
            Thread.sleep(waitTimeMilliSecs);
            // Double the wait time for the next iteration
            waitTimeMilliSecs *= 2;
            if (poll++ > MAX_NUM_POLL_REQUESTS) {
                throw new Exception("Request timed out.  If this is a large set " +
                "of metadata components, check that the time allowed " +
                "by MAX_NUM_POLL_REQUESTS is sufficient.");
            }
            result = metadataConnection.checkRetrieveStatus(
                    asyncResultId, true);
            System.out.println("Retrieve Status: " + result.getStatus());
        } while (!result.isDone());

        if (result.getStatus() == RetrieveStatus.Failed) {
            throw new Exception(result.getErrorStatusCode() + " msg: " +
                    result.getErrorMessage());
        } else if (result.getStatus() == RetrieveStatus.Succeeded) {
            // Print out any warning messages
            StringBuilder buf = new StringBuilder();
            if (result.getMessages() != null) {
                for (RetrieveMessage rm : result.getMessages()) {
                    buf.append(rm.getFileName() + " - " + rm.getProblem());
                }
            }
            if (buf.length() > 0) {
                System.out.println("Retrieve warnings:\n" + buf);
            }

            // Write the zip to the file system
```

```
        System.out.println("Writing results to zip file");
        ByteArrayInputStream bais = new ByteArrayInputStream(result.getZipFile());
        File resultsFile = new File("retrieveResults.zip");
        FileOutputStream os = new FileOutputStream(resultsFile);
        try {
            ReadableByteChannel src = Channels.newChannel(bais);
            FileChannel dest = os.getChannel();
            copy(src, dest);

           System.out.println("Results written to " + resultsFile.getAbsolutePath());

        } finally {
            os.close();
        }
    }
}

/**
 * Helper method to copy from a readable channel to a writable channel,
 * using an in-memory buffer.
 */
private void copy(ReadableByteChannel src, WritableByteChannel dest)
    throws IOException
{
    // Use an in-memory byte buffer
    ByteBuffer buffer = ByteBuffer.allocate(8092);
    while (src.read(buffer) != -1) {
        buffer.flip();
        while(buffer.hasRemaining()) {
            dest.write(buffer);
        }
        buffer.clear();
    }
}

private void setUnpackaged(RetrieveRequest request) throws Exception
{
    // Edit the path, if necessary, if your package.xml file is located elsewhere
    File unpackedManifest = new File(MANIFEST_FILE);
    System.out.println("Manifest file: " + unpackedManifest.getAbsolutePath());

    if (!unpackedManifest.exists() || !unpackedManifest.isFile())
        throw new Exception("Should provide a valid retrieve manifest " +
                "for unpackaged content. " +
                "Looking for " + unpackedManifest.getAbsolutePath());

    // Note that we populate the _package object by parsing a manifest file here.
    // You could populate the _package based on any source for your
    // particular application.
    com.sforce.soap.metadata.Package p = parsePackage(unpackedManifest);
    request.setUnpackaged(p);
}

private com.sforce.soap.metadata.Package parsePackage(File file) throws Exception {
```

```
        try {
            InputStream is = new FileInputStream(file);
            List<PackageTypeMembers> pd = new ArrayList<PackageTypeMembers>();
            DocumentBuilder db =
                DocumentBuilderFactory.newInstance().newDocumentBuilder();
            Element d = db.parse(is).getDocumentElement();
            for (Node c = d.getFirstChild(); c != null; c = c.getNextSibling()) {
                if (c instanceof Element) {
                    Element ce = (Element)c;
                    //
                    NodeList namee = ce.getElementsByTagName("name");
                    if (namee.getLength() == 0) {
                        // not
                        continue;
                    }
                    String name = namee.item(0).getTextContent();
                    NodeList m = ce.getElementsByTagName("members");
                    List<String> members = new ArrayList<String>();
                    for (int i = 0; i < m.getLength(); i++) {
                        Node mm = m.item(i);
                        members.add(mm.getTextContent());
                    }
                    PackageTypeMembers pdi = new PackageTypeMembers();
                    pdi.setName(name);
                    pdi.setMembers(members.toArray(new String[members.size()]));
                    pd.add(pdi);
                }
            }
            com.sforce.soap.metadata.Package r = new com.sforce.soap.metadata.Package();
            r.setTypes(pd.toArray(new PackageTypeMembers[pd.size()]));
            r.setVersion(API_VERSION + "");
            return r;
        } catch (ParserConfigurationException pce) {
            throw new Exception("Cannot create XML parser", pce);
        } catch (IOException ioe) {
            throw new Exception(ioe);
        } catch (SAXException se) {
            throw new Exception(se);
        }
    }


    private void createMetadataConnection(final String username,
            final String password, final String loginUrl)
            throws ConnectionException {

        final ConnectorConfig loginConfig = new ConnectorConfig();
        loginConfig.setAuthEndpoint(loginUrl);
        loginConfig.setServiceEndpoint(loginUrl);
        loginConfig.setManualLogin(true);
        LoginResult loginResult = (new EnterpriseConnection(loginConfig)).login(
                username, password);

        final ConnectorConfig metadataConfig = new ConnectorConfig();
```

```
        metadataConfig.setServiceEndpoint(loginResult.getMetadataServerUrl());
        metadataConfig.setSessionId(loginResult.getSessionId());
        this.metadataConnection = new MetadataConnection(metadataConfig);
    }

    //The sample client application retrieves the user's login credentials.
    // Helper function for retrieving user input from the console
    String getUserInput(String prompt) {
        System.out.print(prompt);
        try {
            return rdr.readLine();
        }
        catch (IOException ex) {
            return null;
        }
    }

}
```

RetrieveRequest

The `RetrieveRequest` parameter specified on a `retrieve()` call encapsulates options for determining which packages or files are retrieved.

The `RetrieveRequest` object consists of the following properties:

Name	Type	Description
`apiVersion`	double	Required. The API version for the retrieve request. The API version determines the fields retrieved for each metadata type. For example, an `icon` field was added to the `CustomTab` `on page 757` for API version 14.0. If you retrieve components for version 13.0 or earlier, the components don't include the `icon` field. In API version 31.0 and later, the API version that's specified in `package.xml` is used for the `retrieve()` call and overrides the version in the `apiVersion` field. If the version isn't specified in `package.xml`, the version in this field is used.
`packageNames`	string[]	A list of package names to be retrieved. If you're retrieving only unpackaged components, don't specify a name here. You can retrieve packaged and unpackaged components in the same retrieve.
`rootTypesWithDependencies`	string[]	A list of component types to retrieve dependencies for. Currently, the only allowed value for this parameter is `Bot`. Use this parameter if any requested metadata components are of type `Bot`.

Name	Type	Description
		Make up to 25 retrieve requests using this parameter per day. A single retrieve request using this parameter can request dependencies for up to 100 components. This field is available in API version 64.0 and later.
`singlePackage`	boolean	Specifies whether only a single package is being retrieved (`true`) or not (`false`). If `false`, then more than one package is being retrieved.
`specificFiles`	string[]	A list of file names to be retrieved. If a value is specified for this property, `packageNames` must be set to `null` and `singlePackage` must be set to `true`.
`unpackaged`	Package	A list of components to retrieve that aren't in a package.

`checkRetrieveStatus()`

Checks the status of the declarative metadata call `checkRetrieveStatus()` and returns the zip file contents.

Syntax

In API version 34.0 and later:

```
RetrieveResult = metadatabinding.checkRetrieveStatus(ID id, boolean includeZip);
```

In API version 33.0 and earlier:

```
RetrieveResult = metadatabinding.checkRetrieveStatus(ID id);
```

Usage

Use `checkRetrieveStatus()` to check the progress of the metadata `retrieve()` operation. The RetrieveResult object that this method returns indicates when the asynchronous `retrieve()` call is completed. If the retrieval is completed, RetrieveResult contains the zip file contents by default. Use the following process to retrieve metadata components with the `retrieve()` call.

1. Issue a `retrieve()` call to start the asynchronous retrieval. An AsyncResult object is returned. Note the value in the `id` field, and use it for the next step.
2. Issue a checkRetrieveStatus() call, and pass in the `id` value from the AsyncResult object from the first step. Check the value of the `done` field of the returned RetrieveResult. If it's `true`, the call is completed, and you proceed to the next step. Otherwise, repeat this step to call checkRetrieveStatus() again until the `done` field is `true`.
3. Retrieve the zip file (`zipFile`) field and other desired fields from RetrieveResult, which the final call to checkRetrieveStatus() returned in the previous step.

In API version 31.0 and later, the process of making a `retrieve()` call has been simplified. You no longer have to call checkStatus() after a `retrieve()` call to obtain the status of the retrieve operation. Instead, make calls to checkRetrieveStatus() only. If the retrieve operation is in progress, call checkRetrieveStatus() again until the retrieve operation is completed. The checkStatus() call is still supported in versions API version 30.0 or earlier, but isn't available in API version 31.0 and later.

Retrieving the Zip File in a Second Process

By default, `checkRetrieveStatus()` returns the zip file on the last call to this operation when the retrieval is completed (`RetrieveResult.isDone() == true`) and then deletes the zip file from the server. Subsequent calls to `checkRetrieveStatus()` for the same retrieve operation can't retrieve the zip file after it has been deleted. Starting with API version 34.0, pass a boolean value for the `includeZip` argument of `checkRetrieveStatus()` to indicate whether to retrieve the zip file. The `includeZip` argument gives you the option to retrieve the file in a separate process after the retrieval operation is completed. For example, a service polls the retrieval status by calling `checkRetrieveStatus(id, false)` in a loop. This call returns the status of the retrieval operation, but doesn't retrieve the zip file. After the retrieval operation is completed, another process, such as a background file transfer service, calls `checkRetrieveStatus(id, true)` to retrieve the zip file. This last call causes the zip file to be deleted from the server.

```
// First process: Poll the retrieval but don't retrieve the zip file.
AsyncResult asyncResult = metadataConnection.retrieve(retrieveRequest);
String asyncResultId = asyncResult.getId();
// Wait for the retrieve to complete
int poll = 0;
long waitTimeMilliSecs = ONE_SECOND;
RetrieveResult result = null;
do {
    Thread.sleep(waitTimeMilliSecs);
    // Check the status but don't retrieve zip file.
    result = metadataConnection.checkRetrieveStatus(asyncResultId, false);
} while (!result.isDone());

// Second process: Retrieve the zip file.
// For example, this process can be a background file transfer service.
// Retrieve the zip file.
result = metadataConnection.checkRetrieveStatus(asyncResultId, true);
// Get the zip file from the RetrieveResult (result) variable
if (result.getStatus() == RetrieveStatus.Succeeded) {
    ByteArrayInputStream bais = new ByteArrayInputStream(result.getZipFile());
    // ...
}
```

Sample Code—Java

See the `retrieve()` sample code for sample usage of this call.

Arguments

Name	Type	Description
`id`	ID	ID obtained from an AsyncResult object returned by a retrieve() call or a subsequent RetrieveResult object returned by a checkRetrieveStatus() call.
`includeZip`	boolean	Set to `true` to retrieve the zip file. You can retrieve the zip file only after the retrieval operation is completed. After the zip file is retrieved, it's deleted from the server. Set to `false` to check the status of the retrieval without attempting to retrieve the zip file. If set to `null`, this argument defaults to `true`, which means that the zip file is retrieved on the last call to `checkRetrieveStatus()` when the retrieval has finished.

Name	Type	Description
		This argument is available in API version 34.0 and later.

Response

RetrieveResult

CHAPTER 9 CRUD-Based Calls

Use CRUD-based calls to work with metadata components in a manner similar to how synchronous API calls in the enterprise WSDL act upon objects.

createMetadata()
Adds one or more new metadata components to your organization synchronously.

readMetadata()
Returns one or more metadata components from your organization synchronously.

updateMetadata()
Updates one or more metadata components in your organization synchronously.

upsertMetadata()
Creates or updates one or more metadata components in your organization synchronously.

deleteMetadata()
Deletes one or more metadata components from your organization synchronously.

renameMetadata()
Renames a metadata component in your organization synchronously.

create()
Deprecated. Adds one or more new metadata components to your organization asynchronously. This call is removed as of API version 31.0 and is available in earlier versions only. Use `createMetadata()` instead.

delete()
Deprecated. Deletes one or more components from your organization asynchronously. This call is removed as of API version 31.0 and is available in earlier versions only. Use `deleteMetadata()` instead.

update()
Deprecated. Updates one or more components in your organization asynchronously. This call is removed as of API version 31.0 and is available in earlier versions only. Use `updateMetadata()` or `renameMetadata()` instead.

`createMetadata()`

Adds one or more new metadata components to your organization synchronously.

Syntax

```
SaveResult[] = metadatabinding.createMetadata(
                Metadata[] metadata);
```

Usage

Use the `createMetadata()` call to create any component that extends Metadata. All components must be of the same type in the same call. For more details, see Metadata Components and Types.

This call executes synchronously, which means that the call returns only when the operation completes.

Starting in API version 34.0, this call supports the AllOrNoneHeader header. By default, if `AllOrNoneHeader` isn't used in API version 34.0 and later, this call can save a partial set of records for records with no errors (equivalent to `AllOrNoneHeader=false`). In API version 33.0 and earlier, the default behavior is to only save all records when there are no failures in any record in the call (equivalent to `AllOrNoneHeader=true`).

Version

Available in API version 30.0 and later.

Permissions

Your client application must be logged in with the Modify Metadata Through Metadata API Functions or Modify All Data permission.

Note: If a user requires access to metadata but not to data, enable the Modify Metadata Through Metadata API Functions on page 7 permission. Otherwise, enable the Modify All Data permission.

Required Fields

The metadata components being created determine required fields. For more information about specific component types, see Metadata Components and Types.

Valid Data Values

You must supply values that are valid for the field's data type, such as integers for integer fields (not alphabetic characters). In your client application, follow the data formatting rules specified for your programming language and development tool. (Your development tool handles the appropriate mapping of data types in SOAP messages.)

String Values

When storing values in string fields, the API trims any leading and trailing whitespace. For example, if the value of a `label` field is entered as `"MyObject "`, the value is stored in the database as `"MyObject"`.

Basic Steps for Creating Metadata Components

Follow this process to create metadata components.

1. Design an array, and populate it with the components that you want to create. All components must be of the same type.
2. Call `createMetadata()` with the component array passed in as an argument.
3. A `SaveResult` object is returned for each component you tried to create. It contains information about whether the operation was successful, the name of the component created, and any errors returned if the operation wasn't successful.

Sample Code—Java

```
public void createCustomObjectSync() {
    try {
        CustomObject co = new CustomObject();
        String name = "MyCustomObject1";
        co.setFullName(name + "__c");
        co.setDeploymentStatus(DeploymentStatus.Deployed);
        co.setDescription("Created by the Metadata API");
        co.setEnableActivities(true);
        co.setLabel(name + " Object");
        co.setPluralLabel(co.getLabel() + "s");
        co.setSharingModel(SharingModel.ReadWrite);

        CustomField nf = new CustomField();
        nf.setType(FieldType.Text);
        nf.setLabel(co.getFullName() + " Name");
        co.setNameField(nf);

        SaveResult[] results = metadataConnection
                .createMetadata(new Metadata[] { co });

        for (SaveResult r : results) {
            if (r.isSuccess()) {
                System.out.println("Created component: " + r.getFullName());
            } else {
                System.out
                        .println("Errors were encountered while creating "
                                + r.getFullName());
                for (Error e : r.getErrors()) {
                    System.out.println("Error message: " + e.getMessage());
                    System.out.println("Status code: " + e.getStatusCode());
                }
            }
        }
    } catch (ConnectionException ce) {
        ce.printStackTrace();
    }
}
```

Arguments

Name	Type	Description
metadata	Metadata[]	Array of one or more metadata components. Limit: 10. (For CustomMetadata on page 656 and CustomApplication on page 623 only, the limit is 200.) You must submit arrays of only one type of component. For example, you can submit an array of 10 custom objects or 10 profiles, but not a mix of both types.

Response

SaveResult[]

readMetadata()

Returns one or more metadata components from your organization synchronously.

Syntax

```
ReadResult = metadataConnection.readMetadata(string metadataType, string[] fullNames);
```

Usage

Use the `readMetadata()` call to retrieve any component that extends Metadata. All components must be of the same type in the same call. For more details, see Metadata Components and Types.

This call executes synchronously, which means that the call returns only when the operation completes.

Version

Available in API version 30.0 and later.

Permissions

Your client application must be logged in with the Modify Metadata Through Metadata API Functions or Modify All Data permission.

Note: If a user requires access to metadata but not to data, enable the Modify Metadata Through Metadata API Functions on page 7 permission. Otherwise, enable the Modify All Data permission.

Basic Steps for Reading Metadata Components

Use the following process to read metadata components:

1. Determine the metadata type of the components you want to read and the `fullName` of each component to read.

   The full names must match one or more full names returned by the `listMetadata()` call, which includes namespace prefixes. If you obtain the `fullName` from a package manifest file, and the component has a namespace prefix, prepend the namespace prefix to the `fullName`. Use this syntax: ***`namespacePrefix__ComponentName`***. For example, for the custom object component `MyCustomObject__c` and the namespace `MyNS`, the `fullName` is `MyNS__MyCustomObject__c`. For more information about the `fullName` field, see Metadata.

   You can read only components of the same type in a single call.

2. Invoke the `readMetadata()` call. For the first argument, pass in the name of the metadata type. The metadata type must match one of the values returned by the `describeMetadata()` call. For the second argument, pass in an array of full names corresponding to the components you wish to get.
3. A `ReadResult` is returned that contains an array of `Metadata` components. Cast each returned `Metadata` object to the metadata type you specified in the call to get the component's properties.

Sample Code—Java

```
public void readCustomObjectSync() {
    try {
        ReadResult readResult = metadataConnection
                .readMetadata("CustomObject", new String[] {
                        "MyCustomObject1__c", "MyCustomObject2__c" });
        Metadata[] mdInfo = readResult.getRecords();
        System.out.println("Number of component info returned: "
                + mdInfo.length);
        for (Metadata md : mdInfo) {
            if (md != null) {
                CustomObject obj = (CustomObject) md;
                System.out.println("Custom object full name: "
                        + obj.getFullName());
                System.out.println("Label: " + obj.getLabel());
                System.out.println("Number of custom fields: "
                        + obj.getFields().length);
                System.out.println("Sharing model: "
                        + obj.getSharingModel());
            } else {
                System.out.println("Empty metadata.");
            }
        }
    } catch (ConnectionException ce) {
        ce.printStackTrace();
    }
}
```

Arguments

Name	Type	Description
`metadataType`	string	The metadata type of the components to read.
`fullNames`	string[]	Array of full names of the components to read. Limit: 10. (For CustomMetadata on page 656 and CustomApplication on page 623 only, the limit is 200.) You must submit arrays of only one type of component. For example, you can submit an array of 10 custom objects or 10 profiles, but not a mix of both types.

Response

ReadResult

`updateMetadata()`

Updates one or more metadata components in your organization synchronously.

Syntax

```
SaveResult[] = metadataConnection.updateMetadata(Metadata[] metadata);
```

Usage

Use the `updateMetadata()` call to update any component that extends Metadata. All components must be of the same type in the same call. For more details, see Metadata Components and Types.

This call executes synchronously, which means that the call returns only when the operation completes.

Starting in API version 34.0, this call supports the AllOrNoneHeader header. By default, if `AllOrNoneHeader` isn't used in API version 34.0 and later, this call can save a partial set of records for records with no errors (equivalent to `AllOrNoneHeader=false`). In API version 33.0 and earlier, the default behavior is to only save all records when there are no failures in any record in the call (equivalent to `AllOrNoneHeader=true`).

Version

Available in API version 30.0 and later.

Permissions

Your client application must be logged in with the Modify Metadata Through Metadata API Functions or Modify All Data permission.

Note: If a user requires access to metadata but not to data, enable the Modify Metadata Through Metadata API Functions on page 7 permission. Otherwise, enable the Modify All Data permission.

Required Fields

You must supply values for all the required fields in the component.

Valid Field Values

You must supply values that are valid for the field's data type, such as integers for integer fields (not alphabetic characters). In your client application, follow the data formatting rules specified for your programming language and development tool. (Your development tool handles the appropriate mapping of data types in SOAP messages.)

String Values

When storing values in string fields, the API trims any leading and trailing white space. For example, if the value of a `label` field is entered as `"MyObject "`, the value is stored in the database as `"MyObject"`.

Basic Steps for Updating Metadata Components

Use this process to update metadata components.

1. Create an array of the components you want to update. All components must be of the same type.
2. Invoke the `updateMetadata()` call, passing in the array of metadata components to update.

A `SaveResult` object is returned for each component you try to update. It contains information about whether the operation was successful, the name of the component updated, and any errors returned if the operation wasn't successful.

Sample Code—Java

```
public void updateCustomObjectSync() {
    try {
        CustomObject co = new CustomObject();
        String name = "MyCustomObject1";
        co.setFullName(name + "__c");
        co.setDeploymentStatus(DeploymentStatus.Deployed);
        co.setDescription("Updated description");
        co.setLabel(name + " Object Update");
        co.setPluralLabel(co.getLabel() + "s");
        co.setSharingModel(SharingModel.ReadWrite);

        // Name field with a type and label is required
        CustomField cf = new CustomField();
        cf.setType(FieldType.Text);
        cf.setLabel(co.getFullName() + " Name");
        co.setNameField(cf);

        SaveResult[] results = metadataConnection
                .updateMetadata(new Metadata[] { co });

        for (SaveResult r : results) {
            if (r.isSuccess()) {
                System.out.println("Updated component: " + r.getFullName());
            } else {
                System.out
                        .println("Errors were encountered while updating "
                                + r.getFullName());
                for (Error e : r.getErrors()) {
                    System.out.println("Error message: " + e.getMessage());
                    System.out.println("Status code: " + e.getStatusCode());
                }
            }
        }
    } catch (ConnectionException ce) {
        ce.printStackTrace();
    }
}
```

Arguments

Name	Type	Description
`metadata`	Metadata[]	Array of one or more metadata components you wish to update. Limit: 10. (For CustomMetadata on page 656 and CustomApplication on page 623 only, the limit is 200.)

Name	Type	Description
		You must submit arrays of only one type of component. For example, you can submit an array of 10 custom objects or 10 profiles, but not a mix of both types.

Response

SaveResult[]

upsertMetadata()

Creates or updates one or more metadata components in your organization synchronously.

Syntax

```
UpsertResult[] = metadataConnection.upsertMetadata(Metadata[] metadata);
```

Usage

Use the `upsertMetadata()` call to create or update any component that extends Metadata. All components must be of the same type in the same call. For more details, see Metadata Components and Types.

If the specified components currently exist in your organization, the `upsertMetadata()` call updates them. Otherwise, `upsertMetadata()` creates these components. The `fullname` field matches the components. This call executes synchronously, which means that the call returns only after the operation is completed.

Starting in API version 34.0, this call supports the AllOrNoneHeader header. By default, if `AllOrNoneHeader` isn't used in API version 34.0 and later, this call can save a partial set of records for records with no errors (equivalent to `AllOrNoneHeader=false`). In API version 33.0 and earlier, the default behavior is to only save all records when there are no failures in any record in the call (equivalent to `AllOrNoneHeader=true`).

Version

Available in API version 31.0 and later.

Permissions

Your client application must be logged in with the Modify Metadata Through Metadata API Functions or Modify All Data permission.

Note: If a user requires access to metadata but not to data, enable the Modify Metadata Through Metadata API Functions on page 7 permission. Otherwise, enable the Modify All Data permission.

Required Fields

You must supply values for all the required fields in the component.

Valid Field Values

You must supply values that are valid for the field's data type, such as integers (not alphabetic characters) for integer fields. In your client application, follow the data formatting rules that are specified for your programming language and development tool. (Your development tool handles the appropriate mapping of data types in SOAP messages.)

String Values

The API trims any leading and trailing white space when storing values in string fields. For example, if the value of a `label` field is entered as `" MyObject "`, the value is stored in the database as `"MyObject"`.

Basic Steps for Upserting Metadata Components

Use this process to upsert metadata components.

1. Create an array of Metadata objects that correspond to the components that you want to create or update. All components must be of the same type.
2. Invoke `upsertMetadata()`, passing in the array of metadata components that you created in the previous step.

   The `upsertMetadata()` call returns an array of `UpsertResult` objects. Each returned `UpsertResult` corresponds to a component that you upserted and contains information about the upsert operation—whether the operation was successful, the name of the component that was upserted, a flag indicating whether the component was created, and any errors that were returned if the operation wasn't successful.

Sample Code—Java

```
public void upsertMetadataSample() {
    try {
        // Create custom object to upsert
        CustomObject co = new CustomObject();
        String name = "MyCustomObject";
        co.setFullName(name + "__c");
        co.setDeploymentStatus(DeploymentStatus.Deployed);
        co.setDescription("Upserted by the Metadata API");
        co.setEnableActivities(true);
        co.setLabel(name + " Object");
        co.setPluralLabel(co.getLabel() + "s");
        co.setSharingModel(SharingModel.ReadWrite);

        CustomField nf = new CustomField();
        nf.setType(FieldType.Text);
        nf.setLabel("CustomField1");
        co.setNameField(nf);

        // Upsert the custom object
        UpsertResult[] results = metadataConnection
                .upsertMetadata(new Metadata[] { co });

        for (UpsertResult r : results) {
            if (r.isSuccess()) {
                System.out.println("Success!");
```

```
                if (r.isCreated()) {
                    System.out.println("Created component: "
                            + r.getFullName());
                } else {
                    System.out.println("Updated component: "
                            + r.getFullName());
                }
            } else {
                System.out
                .println("Errors were encountered while upserting "
                        + r.getFullName());
                for (Error e : r.getErrors()) {
                    System.out.println("Error message: " + e.getMessage());
                    System.out.println("Status code: " + e.getStatusCode());
                }
            }
        }
    } catch (ConnectionException ce) {
        ce.printStackTrace();
    }
}
```

Arguments

Name	Type	Description
`metadata`	Metadata[]	An array of one or more metadata components that you want to create or update Limit: 10. You must submit arrays of only one type of component. For example, you can submit an array of 10 custom objects or 10 profiles, but not a mix of both types.

Response

UpsertResult[]

deleteMetadata()

Deletes one or more metadata components from your organization synchronously.

Syntax

```
DeleteResult[] = metadataConnection.delete(string metadataType, string[] fullNames);
```

Usage

Use the `deleteMetadata()` call to delete any component that extends Metadata. All components must be of the same type in the same call. For more details, see Metadata Components and Types.

This call executes synchronously, which means that the call returns only when the operation completes.

Starting in API version 34.0, this call supports the AllOrNoneHeader header. By default, if the `AllOrNoneHeader` isn't used in any API version, this call can delete a partial set of records for records with no errors (equivalent to `AllOrNoneHeader=false`). If `AllOrNoneHeader` is set to `true`, no records are deleted if one or more records cause a failure.

Version

Available in API version 30.0 and later.

Permissions

Your client application must be logged in with the Modify Metadata Through Metadata API Functions or Modify All Data permission.

Note: If a user requires access to metadata but not to data, enable the Modify Metadata Through Metadata API Functions on page 7 permission. Otherwise, enable the Modify All Data permission.

Rules and Guidelines

When deleting components, consider the following rules and guidelines:

- Your client application must be logged in with sufficient access rights to delete individual components within the specified component. For more information, see Factors that Affect Data Access in the *Salesforce Object Reference*.
- In addition, sometimes you also need permission to access this component's parent component.
- To ensure referential integrity, this call supports cascading deletions. If you delete a parent component, you delete its children automatically, as long as each child component can be deleted.

Basic Steps for Deleting Metadata Components

Use the following process to delete metadata components:

1. Determine the metadata type of the components you want to delete and the `fullName` of each component to delete. You can delete only components of the same type in a single call. The full names must match one or more full names returned by the `listMetadata()` call, which includes namespace prefixes. If you obtain the `fullName` from a package manifest file, and the component has a namespace prefix, prepend the namespace prefix to the `fullName`. Use this syntax: ***namespacePrefix__ComponentName***. For example, for the custom object component `MyCustomObject__c` and the namespace `MyNS`, the `fullName` is `MyNS__MyCustomObject__c`. See Metadata for more details on the `fullName` field.
2. Invoke the `deleteMetadata()` call. For the first argument, pass in the name of the metadata type. For the second argument, pass in an array of full names corresponding to the components you wish to delete.

   A `DeleteResult` object is returned for each component you try to delete. It contains information about whether the operation was successful, the name of the deleted component, and any errors returned if the operation wasn't successful.

Sample Code—Java

```
public void deleteCustomObjectSync() {
    try {
        DeleteResult[] results = metadataConnection.deleteMetadata(
                "CustomObject", new String[] { "MyCustomObject1__c",
                        "MyCustomObject2__c" });
        for (DeleteResult r : results) {
            if (r.isSuccess()) {
                System.out.println("Deleted component: " + r.getFullName());
            } else {
                System.out
                        .println("Errors were encountered while deleting "
                                + r.getFullName());
                for (Error e : r.getErrors()) {
                    System.out.println("Error message: " + e.getMessage());
                    System.out.println("Status code: " + e.getStatusCode());
                }
            }
        }
    } catch (ConnectionException ce) {
        ce.printStackTrace();
    }
}
```

Arguments

Name	Type	Description
`metadataType`	string	The metadata type of the components to delete.
`fullNames`	string[]	Array of full names of the components to delete. Limit: 10. (For CustomMetadata on page 656 and CustomApplication on page 623 only, the limit is 200.) Submit arrays of only one type of component. For example, you can submit an array of 10 custom objects or 10 profiles, but not a mix of both types.

Response

DeleteResult[]

`renameMetadata()`

Renames a metadata component in your organization synchronously.

Syntax

```
SaveResult = metadataConnection.renameMetadata(string metadataType, String oldFullname,
String newFullname);
```

Usage

Use the `renameMetadata()` call to rename one metadata component in your organization. This call executes synchronously, meaning the call returns only when the operation completes.

You can use this call to rename any of the objects that extend Metadata. For more details, see Metadata Components and Types.

Version

Available in API version 30.0 and later.

Permissions

Your client application must be logged in with the Modify Metadata Through Metadata API Functions or Modify All Data permission.

Note: If a user requires access to metadata but not to data, enable the Modify Metadata Through Metadata API Functions on page 7 permission. Otherwise, enable the Modify All Data permission.

Basic Steps for Renaming Metadata Components

Use this process to rename a metadata component.

1. Determine the metadata type of the component you want to rename, its current full name, and the new full name. See Metadata for more details on the `fullName` field.
2. Invoke the `renameMetadata()` call. For the first argument, pass in the name of the metadata type. Pass in the old full name as the second argument and the new full name as the last argument.

   A SaveResult object that contains information about whether the operation was successful is returned. If successful, the object contains the name of the renamed component, which is the new name. If it wasn't successful, the object returns any errors.

Sample Code—Java

```
public void renameCustomObjectSync() {
 try {
    SaveResult[] results = metadataConnection.renameMetadata(
        "CustomObject", "MyCustomObject1__c","MyCustomObject1New__c");
    for (SaveResult r : results) {
     if (r.isSuccess()) {
      System.out.println("Renamed component: " + r.getName());
     }
     else {
      System.out.println("Errors were encountered while renaming " + r.getName());
      for(Error e : r.getErrors()) {
       System.out.println("Error message: " + e.getMessage());
       System.out.println("Status code: " + e.getStatusCode());
```

```
      }
     }
    }
  } catch (ConnectionException ce) {
    ce.printStackTrace();
  } catch (InterruptedException ie) {
    ie.printStackTrace();
  }
}
```

Arguments

Name	Type	Description
`metadataType`	string	The metadata type of the components to rename.
`oldFullName`	string	The current component full name.
`newFullName`	string	The full name of the new component.

Response

SaveResult

`create()`

Deprecated. Adds one or more new metadata components to your organization asynchronously. This call is removed as of API version 31.0 and is available in earlier versions only. Use `createMetadata()` instead.

Syntax

```
AsyncResult[] = metadatabinding.create(Metadata[] metadata);
```

Usage

Use this call to add one or more metadata components to your organization.

Version

This call is available in API version 30.0 and earlier only. This call isn't available in API version 31.0 and later. Use createMetadata() instead.

Permissions

Your client application must be logged in with the Modify Metadata Through Metadata API Functions or Modify All Data permission.

Note: If a user requires access to metadata but not to data, enable the Modify Metadata Through Metadata API Functions on page 7 permission. Otherwise, enable the Modify All Data permission.

Required Fields

The metadata components being created determine required fields. For more information about specific component types, see Metadata Components and Types on page 158.

Valid Data Values

You must supply values that are valid for the field's data type, such as integers for integer fields (not alphabetic characters). In your client application, follow the data formatting rules specified for your programming language and development tool. (Your development tool handles the appropriate mapping of data types in SOAP messages).

String Values

When storing values in string fields, the API trims any leading and trailing whitespace. For example, if the value of a `label` field is entered as `"MyObject "`, the value is stored in the database as `"MyObject"`.

Basic Steps for Creating Metadata Components

Use the following process to create metadata components:

1. Design an array, and populate it with the components you want to create. All components must be of the same type.
2. Call create() with the component array passed in as an argument.
3. An AsyncResult object is returned for each component you try to create, and is updated with status information as the operation moves from a queue to completed or error state. Call checkStatus() in a loop until the status values in AsyncResult indicate that all create operations are completed. Start with a wait time of 1 second between iterations of checkStatus() calls, and double the wait time each time you make a subsequent call.

Sample Code—Java

See Step 3: Walk Through the Java Sample Code on page 16 for sample Java code using the `create()` call.

Arguments

Name	Type	Description
`metadata`	Metadata[]	Array of one or more metadata components. Limit: 10. You must submit arrays of only one type of component. For example, you could submit an array of 10 custom objects or 10 profiles, but not a mix of both types.

Response

AsyncResult[]

SEE ALSO:

- createMetadata()
- update()
- delete()
- checkStatus()

delete()

Deprecated. Deletes one or more components from your organization asynchronously. This call is removed as of API version 31.0 and is available in earlier versions only. Use `deleteMetadata()` instead.

You can use this call to delete any of the objects that extend Metadata. For more details, see Metadata Components and Types on page 158.

Syntax

```
AsyncResult[] = metadataConnection.delete(Metadata[] metadata);
```

Usage

Use this call to delete one or more components from your organization.

Version

This call is available in API version 30.0 and earlier only. This call isn't available in API version 31.0 and later. Use deleteMetadata() instead.

Permissions

Your client application must be logged in with the Modify Metadata Through Metadata API Functions or Modify All Data permission.

Note: If a user requires access to metadata but not to data, enable the Modify Metadata Through Metadata API Functions on page 7 permission. Otherwise, enable the Modify All Data permission.

Rules and Guidelines

When deleting components, consider the following rules and guidelines:

- Your client application must be logged in with sufficient access rights to delete individual components within the specified component. For more information, see Factors that Affect Data Access in the *Salesforce Object Reference*.
- In addition, sometimes you also need permission to access this component's parent component.
- To ensure referential integrity, this call supports cascading deletions. If you delete a parent component, you delete its children automatically, as long as each child component can be deleted.

Basic Steps for Deleting Metadata Components

Use the following process to delete metadata components:

1. Determine the `fullName` of each component you want to delete. See Metadata for more details on the `fullName` field. You can only delete components of the same type in a single call.
2. Invoke the `delete()` call, passing in the array of metadata components with `fullName` specified.
3. An AsyncResult object is returned for each component you try to delete, and is updated with status information as the operation moves from a queue to completed or error state. Call checkStatus() in a loop until the status values in AsyncResult indicate that all the delete operations are completed. Start with a wait time of 1 second between iterations of checkStatus() calls, and double the wait time each time you make a subsequent call.

Sample Code—Java

```
public void deleteCustomObject() {
  try {
    CustomObject co = new CustomObject();
    co.setFullName("MyCustomObject__c");
    AsyncResult[] ars = metadataConnection.create(new Metadata[]
       {co});
    AsyncResult asyncResult = ars[0];
    long waitTimeMilliSecs = 1000;
    while (!asyncResult.isDone()) {
      Thread.sleep(waitTimeMilliSecs);
      // double the wait time for the next iteration
      waitTimeMilliSecs *= 2;
      asyncResult = mdConnection.checkStatus(
        new String[] {asyncResult.getId()})[0];
      System.out.println("Status is: " + asyncResult.getState());
    }
  } catch (ConnectionException ce) {
    ce.printStackTrace();
  } catch (InterruptedException ie) {
    ie.printStackTrace();
  }
}
```

Arguments

Name	Type	Description
`metadata`	Metadata[]	Array of one or more metadata components. Only set the `fullName` field in the Metadata object. Limit: 10. You must submit arrays of only one type of component. For example, you could submit an array of 10 custom objects or 10 profiles, but not a mix of both types.

Response

AsyncResult[]

SEE ALSO:

- deleteMetadata()
- create()
- update()
- checkStatus()

update()

Deprecated. Updates one or more components in your organization asynchronously. This call is removed as of API version 31.0 and is available in earlier versions only. Use `updateMetadata()` or `renameMetadata()` instead.

This call can be used to update any of the objects that extend Metadata. For more details, see Metadata Components and Types on page 158.

Syntax

```
AsyncResult[] = metadataConnection.update(UpdateMetadata[] metadata);
```

Usage

Use this call to update one or more components. This call is analogous to the `ALTER TABLE` statement in SQL.

Version

This call is available in API version 30.0 and earlier only. This call isn't available in API version 31.0 and later. Use updateMetadata() instead to update metadata components or renameMetadata() to rename a metadata component.

Permissions

Your client application must be logged in with the Modify Metadata Through Metadata API Functions or Modify All Data permission.

Note: If a user requires access to metadata but not to data, enable the Modify Metadata Through Metadata API Functions on page 7 permission. Otherwise, enable the Modify All Data permission.

Required Fields

You must supply values for all the required fields in the component.

Valid Field Values

You must supply values that are valid for the field's data type, such as integers for integer fields (not alphabetic characters). In your client application, follow the data formatting rules specified for your programming language and development tool. (Your development tool handles the appropriate mapping of data types in SOAP messages).

String Values

When storing values in string fields, the API trims any leading and trailing white space. For example, if the value of a `label` field is entered as `"MyObject "`, the value is stored in the database as `"MyObject"`.

Basic Steps for Updating Metadata Components

Use this process to update metadata components:

1. Create an array of `UpdateMetadata` components, and populate it with the components you want to update. All components must be of the same type.
2. Invoke the `update()` call, passing in the array of metadata components to update.
3. An AsyncResult object is returned for each component you try to update, and is updated with status information as the operation moves from a queue to completed or error state. In a loop, call checkStatus() until the status values in AsyncResult indicate that all the update operations are completed. Start with a wait time of 1 second between iterations of checkStatus() calls, and double the wait time each time you make a subsequent call.

Sample Code—Java

```
public void updateCustomObject() {
  try {
    CustomObject co = new CustomObject();
    String name = "MyCustomObject";
    co.setFullName(name + "__c");
    co.setDeploymentStatus(DeploymentStatus.Deployed);
    co.setDescription("Created by the Metadata API");
    co.setEnableActivities(true);
    co.setLabel(name + " Object");
    co.setPluralLabel(co.getLabel() + "s");
    co.setSharingModel(SharingModel.ReadWrite);

    CustomField nf = new CustomField();
    nf.setType(FieldType.Text);
    nf.setLabel(co.getFullName() + " Name");

    co.setNameField(nf);

    UpdateMetadata updateMetadata = new UpdateMetadata();
    updateMetadata.setMetadata(co);
    updateMetadata.setCurrentName("TheCurrentName");

    AsyncResult[] ars = metadataConnection.update(new UpdateMetadata[]
        { updateMetadata });
    AsyncResult asyncResult = ars[0];
```

```
    // set initial wait time to one second in milliseconds
    long waitTimeMilliSecs = 1000;
    while (!asyncResult.isDone()) {
      Thread.sleep(waitTimeMilliSecs);
      // double the wait time for the next iteration
      waitTimeMilliSecs *= 2;
      asyncResult = metadataConnection.checkStatus(
        new String[] {asyncResult.getId()})[0];
      System.out.println("Status is: " + asyncResult.getState());
    }

    if (asyncResult.getState() != AsyncRequestState.Completed) {
      System.out.println(asyncResult.getStatusCode() + " msg: " +
          asyncResult.getMessage());
    }
  } catch (InterruptedException ie) {
    ie.printStackTrace();
  } catch (ConnectionException ce) {
    ce.printStackTrace();
  }
}
```

Arguments

Name	Type	Description
`metadata`	UpdateMetadata[]	Array of one or more UpdateMetadata data structures that represent the components you wish to update. Limit: 10. You must submit arrays of only one type of component. For example, you could submit an array of 10 custom objects or 10 profiles, but not a mix of both types.

UpdateMetadata

One or more UpdateMetadata objects are defined in the `metadata` argument. This object can be used to update any of the objects that extend Metadata. For more details, see Metadata Components and Types on page 158. Each UpdateMetadata object has the following fields:

Field	Field Type	Description
`currentName`	string	The API name of the component or field before the update. For example, if you wanted to update a CustomObject named Foo, the value of this field would be `Foo__c`. This value is supplied because this call can change the name, and the value here provides mapping.
`metadata`	Metadata	Full specification of the component or field you want to update.

Response

AsyncResult[]

SEE ALSO:

- updateMetadata()
- create()
- delete()
- checkStatus()

CHAPTER 10 Utility Calls

Use utility calls to gather information that is useful for working with the file-based or CRUD-based calls.

- (Deprecated) checkStatus()
- describeMetadata()
- describeValueType()
- listMetadata()

`checkStatus()`

Deprecated. Checks the status of asynchronous metadata calls `create()`, `update()`, or `delete()`, or the declarative metadata call `retrieve()`. This call is removed as of API version 31.0 and is available only in earlier versions.

Note: Starting in API version 29.0, you no longer have to call checkStatus() on page 106 after a deploy() on page 56 call to get information about deployments. Similarly, starting in API version 31.0, you no longer have to call checkStatus() on page 106 after a retrieve() on page 75 call. The checkStatus() on page 106 call has been replaced by checkDeployStatus() and checkRetrieveStatus() for deploy and retrieve operations respectively.

Syntax

```
AsyncResult[] = metadatabinding.checkStatus(ID[] ids);
```

Usage

Use this call to check whether an asynchronous metadata call or declarative metadata call has completed.

Version

This call is available only in API version 30.0 and earlier. This call isn't available in API version 31.0 and later.

Sample Code—Java

See Step 3: Walk Through the Java Sample Code on page 16 for sample Java code using this call.

Arguments

Name	Type	Description
ids	ID[]	Array of one or more IDs. Each ID is returned in an AsyncResult and corresponds to a component being created, updated, deleted, deployed, or retrieved.

Response

AsyncResult[]

describeMetadata()

This call retrieves the metadata that describes your organization. This information includes Apex classes and triggers, custom objects, custom fields on standard objects, tab sets that define an app, and many other metadata types.

Syntax

```
DescribeMetadataResult = metadataConnection.describeMetadata(double apiVersion);
```

Arguments

Name	Type	Description
apiVersion	double	The API version for which you want metadata, for example, 64.0.

Permissions

Your client application must be logged in with the Modify Metadata Through Metadata API Functions or Modify All Data permission.

Note: If a user requires access to metadata but not to data, enable the Modify Metadata Through Metadata API Functions on page 7 permission. Otherwise, enable the Modify All Data permission.

Sample Code—Java

```
public void describeMetadata() {
  try {
    double apiVersion = 21.0;
    // Assuming that the SOAP binding has already been established.
    DescribeMetadataResult res =
        metadataConnection.describeMetadata(apiVersion);
    StringBuffer sb = new StringBuffer();
    if (res != null && res.getMetadataObjects().length > 0) {
      for (DescribeMetadataObject obj : res.getMetadataObjects()) {
        sb.append("**********************************************\n");
        sb.append("XMLName: " + obj.getXmlName() + "\n");
```

```
        sb.append("DirName: " + obj.getDirectoryName() + "\n");
        sb.append("Suffix: " + obj.getSuffix() + "\n");
        sb.append("***********************************************\n");
      }
    } else {
      sb.append("Failed to obtain metadata types.");
    }
    System.out.println(sb.toString());
  } catch (ConnectionException ce) {
    ce.printStackTrace();
  }
}
```

Response

DescribeMetadataResult

When to Use `describeMetadata()` and `describeValueType()`?

Use the `describeMetadata()` call to get high-level information about all the metadata types that are available for your organization, such as type names and file suffixes. Use the `describeValueType()` call to get granular information about a specific metadata type, such as fields contained within the type.

`describeValueType()`

Retrieves the metadata describing a given metadata type (value type).

`describeValueType()` accepts a namespace and a type name, and returns a `DescribeValueTypeResult` object. This call is available in API version 33.0 and later.

Syntax

```
DescribeValueTypeResult = connection.describeValueType("{namespace}type_name");
```

Example

Describe Apex class metadata in the Metadata namespace:

```
DescribeValueTypeResult =
metadataConnection.describeValueType("{http://soap.sforce.com/2006/04/metadata}ApexClass");
```

Describe Apex class metadata in the Tooling namespace:

```
DescribeValueTypeResult =
toolingConnection.describeValueType("{urn:metadata.tooling.soap.sforce.com}ApexClass");
```

Arguments

Name	Type	Description
`type`	string	The name of the metadata type for which you want metadata; for example, `ApexClass`. Include the namespace.

Permissions

Your client application must be logged in with the Modify Metadata Through Metadata API Functions or Modify All Data permission.

Note: If a user requires access to metadata but not to data, enable the Modify Metadata Through Metadata API Functions on page 7 permission. Otherwise, enable the Modify All Data permission.

Sample Code—Java

The following example describes several metadata types by specifying the Metadata namespace. Each metadata type is described using the helper method, `doDescribe()`, which calls the `describeValueType()` Metadata API call. The sample retrieves information from the returned `DescribeValueTypeResult`: a property, the parent field (if any), and the fields. Next, the sample iterates through the fields and outputs information about each field.

```
public void describeValueType() throws ConnectionException {
    doDescribe("{http://soap.sforce.com/2006/04/metadata}CustomObject");
    doDescribe("{http://soap.sforce.com/2006/04/metadata}CustomField");
    doDescribe("{http://soap.sforce.com/2006/04/metadata}EmailTemplate");
}

public void doDescribe(String type) throws ConnectionException {
    DescribeValueTypeResult result = metadataConnection.describeValueType(type);
    StringBuffer sb = new StringBuffer();

    sb.append("Describing " + type + " ...\n");

    if (result.getApiCreatable() == true) {
        sb.append("Is API creatable.\n");
    } else {
        sb.append("Is not API creatable.\n");
    }

    ValueTypeField parentField = result.getParentField();
    if (parentField != null) {
        sb.append("** Parent type fields **\n");
        if (parentField.getIsForeignKey()) {
            sb.append("This field is a foreign key.\n");
            for (String fkDomain : parentField.getForeignKeyDomain()) {
                sb.append("Foreign key domain: " + fkDomain + "\n");
            }
        }
    }

    sb.append("** Value type fields **\n");
```

```
    for(ValueTypeField field : result.getValueTypeFields()) {
        sb.append("**************************************************\n");
        sb.append("Name: " + field.getName() + "\n");
        sb.append("SoapType: " + field.getSoapType() + "\n");
        if (field.getIsForeignKey()) {
            sb.append("This field is a foreign key.\n");
            for (String fkDomain : field.getForeignKeyDomain()) {
                sb.append("Foreign key domain: " + fkDomain + "\n");
            }
        }
        sb.append("**************************************************\n");
    }
    System.out.println(sb.toString());
}
```

To run the previous example with the Tooling WSDL, replace the namespace with the Tooling namespace in the helper function call as follows. Also, use the Tooling connection instead of the Metadata connection to make the `describeValueType()` call.

```
    doDescribe("{urn:metadata.tooling.soap.sforce.com}CustomObject");
    doDescribe("{urn:metadata.tooling.soap.sforce.com}CustomField");
    doDescribe("{urn:metadata.tooling.soap.sforce.com}EmailTemplate");
```

After you run the sample, the output looks similar to the following.

```
Describing {http://soap.sforce.com/2006/04/metadata}CustomObject ...
Is API creatable.
** Value type fields **
**************************************************
Name: actionOverrides
SoapType: ActionOverride
**************************************************
**************************************************
Name: allowInChatterGroups
SoapType: boolean
**************************************************
**************************************************
Name: articleTypeChannelDisplay
SoapType: ArticleTypeChannelDisplay
**************************************************
**************************************************
Name: businessProcesses
SoapType: BusinessProcess
**************************************************
**************************************************
Name: compactLayoutAssignment
SoapType: string
**************************************************
**************************************************
Name: compactLayouts
SoapType: CompactLayout
**************************************************
**************************************************
Name: customHelp
SoapType: string
```

```
This field is a foreign key.
Foreign key domain: ApexPage
Foreign key domain: Scontrol
************************************************
<The rest of the output for CustomObject has been omitted for brevity.>

Describing {http://soap.sforce.com/2006/04/metadata}CustomField ...
Is API creatable.
** Parent type fields **
This field is a foreign key.
Foreign key domain: CustomObject
** Value type fields **
************************************************
Name: caseSensitive
SoapType: boolean
************************************************
************************************************

Name: defaultValue
SoapType: string
************************************************

<The rest of the output has been omitted for brevity.>
```

Response

DescribeValueTypeResult

listMetadata()

This call retrieves property information about metadata components in your organization. Data is returned for the components that match the criteria specified in the queries parameter. The queries array can contain up to three `ListMetadataQuery` queries for each call. This call supports every metadata type: both top-level, such as CustomObject and ApexClass, and child types, such as CustomField and RecordType.

Syntax

```
FileProperties[] = metadataConnection.listMetadata(ListMetadataQuery[] queries, double
asOfVersion);
```

Usage

This call is useful when you want to identify individual components in `package.xml` for a retrieve() call or if you want a high-level view of particular metadata types in your organization. For example, you can use this call to return a list of names of all the CustomObject or Layout components in your organization. You can use this information to make a subsequent retrieve() call to return a subset of these components. For more information about working with `package.xml`, see Deploying and Retrieving Metadata on page 25.

This call is synchronous, so the results are returned in one call. This call differs from asynchronous calls, such as retrieve(), where at least one subsequent call is required to get the results.

Permissions

Your client application must be logged in with the Modify Metadata Through Metadata API Functions or Modify All Data permission.

Note: If a user requires access to metadata but not to data, enable the Modify Metadata Through Metadata API Functions on page 7 permission. Otherwise, enable the Modify All Data permission.

Sample Code—Java

This sample code lists information about your custom objects. The code assumes that the SOAP binding has already been established.

```
public void listMetadata() {
  try {
    ListMetadataQuery query = new ListMetadataQuery();
    query.setType("CustomObject");
    //query.setFolder(null);
    double asOfVersion = 64.0;
    // Assuming that the SOAP binding has already been established.
    FileProperties[] lmr = metadataConnection.listMetadata(
        new ListMetadataQuery[] {query}, asOfVersion);
    if (lmr != null) {
      for (FileProperties n : lmr) {
        System.out.println("Component fullName: " + n.getFullName());
        System.out.println("Component type: " + n.getType());
      }
    }
  } catch (ConnectionException ce) {
    ce.printStackTrace();
  }
}
```

Arguments

Name	Type	Description
`queries`	ListMetadataQuery[]	A list of objects that specify which components you're interested in.
`asOfVersion`	double	The API version for the metadata listing request. If you don't specify a value in this field, it defaults to the API version specified when you logged in. This field allows you to override the default and set another API version. For example, you can list the metadata for a metadata type that was added in a later version than the API version specified when you logged in. This field is available in API version 18.0 and later.

Response

FileProperties

`ListMetadataQuery`

The `ListMetadataQuery` parameter represents a list of objects that specify which components you are interested in.

Name	Type	Description
`folder`	string	The folder associated with the component. This field is required for components that use folders, such as Dashboard, Document, EmailTemplate, or Report.
`type`	string	Required. The metadata type, such as `CustomObject`, `CustomField`, or `ApexClass`.

CHAPTER 11 Result Objects

Use the following objects to get the results of your file-based or CRUD-based calls.

AsyncResult
Contains the ID of a deployment or retrieval. In API version 28.0 and earlier, contains status information of any asynchronous metadata call.

CancelDeployResult
Contains information about a deployment cancellation—whether the cancellation completed and the deployment ID.

DeployResult
Contains information about the success or failure of the associated `deploy()` call.

DescribeMetadataResult
Contains information about the organization that is useful for developers working with declarative metadata.

DescribeValueTypeResult
Contains information about a value type that's useful for developers working with declarative metadata.

ReadResult
Contains result information for the `readMetadata` call.

RetrieveResult
The `retrieve()` call returns an array of RetrieveResult objects.

SaveResult
Contains result information for the `createMetadata`, `updateMetadata`, or `renameMetadata` call.

DeleteResult
Contains result information for the `deleteMetadata` call.

UpsertResult
Contains information about the result of the associated `upsertMetadata()` call.

Error
Represents an error that occurred during a synchronous CRUD (`createMetadata()`, `updateMetadata()`, or `deleteMetadata()`) operation.

AsyncResult

Contains the ID of a deployment or retrieval. In API version 28.0 and earlier, contains status information of any asynchronous metadata call.

API Version 31.0 and Later

In API version 31.0, the process of retrieving metadata has been simplified and retrieval properties have been moved to RetrieveResult. Also, the asynchronous create() on page 98, update() on page 102, and delete() on page 100 calls have been removed. Therefore, only the `id` field in AsyncResult is used. The `id` field is the ID of a deployment or retrieval.

These asynchronous calls can return AsyncResult.

- deploy()
- retrieve()

AsyncResult has this field that's in use.

Name	Type	Description
`id`	ID	Required. The ID of the component that's being deployed or retrieved.

All fields in AsyncResult other than `id` are deprecated as of API version 31.0. These fields exist but are no longer in use.

- `done`
- `message`
- `state`
- `statusCode`

API Versions 29.0 and 30.0

In API version 29.0, Salesforce moved several properties from the AsyncResult object to the DeployResult object and added several new ones, to improve the process for getting information about deployments. For more information about these changes, see deploy().

In API versions 29.0 and 30.0, AsyncResult is returned by the same asynchronous calls as in API version 28.0 and earlier, but it has different fields.

Name	Type	Description
`done`	boolean	Required. Indicates whether the call has been completed (`true`) or not (`false`).
`id`	ID	Required. The ID of the component that's being created, updated, deleted, deployed, or retrieved.
`message`	string	The message that corresponds to the returned `statusCode` field, if any.
`state`	AsyncRequestState (enumeration of type string)	Required. The `AsyncRequestState` object has one of four possible values. • `Queued`: This call hasn't started. It's waiting in a queue. • `InProgress`: This call has started but hasn't been completed. • `Completed`: This call has been completed. • `Error`: An error occurred. See the `statusCode` for more information.

Name	Type	Description
`statusCode`	StatusCode (enumeration of type string)	If an error occurred during the `create()`, `update()`, or `delete()` call, a status code is returned, and the message that corresponds to the status code is returned in the `message` field. For a description of each StatusCode value, see StatusCode in the SOAP API Developer Guide.

API Version 28.0 and Earlier

In API version 28.0 and earlier, these asynchronous calls can return AsyncResult.

- create()
- delete()
- deploy()
- retrieve()
- update()

Use the checkStatus() call against each object to discover when the call is completed for that object. Salesforce updates each AsyncResult object as the call is completed or when errors occur.

Similarly, the deploy() and retrieve() calls use AsyncResult, though you must subsequently use checkDeployStatus() or checkRetrieveStatus() respectively to get more status information for the deployment or retrieval.

AsyncResult has the following fields.

Name	Type	Description
`checkOnly`	boolean	Indicates whether this deployment is used to check the validity of the deployed files without changing the organization (`true`) or not (`false`). A check-only deployment doesn't deploy any components or change the organization in any way. This field is available in API version 16.0 and later and is relevant only for the deploy() call.
`done`	boolean	Required. Indicates whether the call has been completed (`true`) or not (`false`).
`id`	ID	Required. The ID of the component that's being created, updated, deleted, deployed, or retrieved.
`message`	string	The message that corresponds to the returned `statusCode` field, if any.
`numberComponentErrors`	int	The number of components that generated errors during this deployment. This field is available in API version 16.0 and later and is relevant only for the deploy() call.
`numberComponentsDeployed`	int	The number of components that have been deployed for this deployment. This field in conjunction with the `numberComponentsTotal` field gives you an indication of the progress of the deployment. This field is available in API version 16.0 and later and is relevant only for the deploy() call.

Name	Type	Description
`numberComponentsTotal`	int	The total number of components in the deployment. This field in conjunction with the `numberComponentsDeployed` field gives you an indication of the progress of the deployment. This field is available in API version 16.0 and later and is relevant only for the deploy() call.
`numberTestErrors`	int	The number of Apex tests that generated errors during this deployment. This field is available in API version 16.0 and later and is relevant only for the deploy() call.
`numberTestsCompleted`	int	The number of Apex tests that have been completed for this deployment. This field in conjunction with the `numberTestsTotal` field gives you an indication of the progress of tests for the deployment. This field is available in API version 16.0 and later and is relevant only for the deploy() call.
`numberTestsTotal`	int	The total number of Apex tests in the deployment. This field in conjunction with the `numberTestsCompleted` field gives you an indication of the progress of tests for the deployment. The value in this field isn't accurate until the deployment has started running tests for the components that are being deployed. This field is available in API version 16.0 and later and is relevant only for the deploy() call.
`secondsToWait`	int	This field is no longer supported for API version 13.0 and later and is provided only for backward compatibility. The field was removed in API version 17.0. Indicates the number of seconds before the call is likely to be completed. This number is an estimate only. A reasonable approach is to wait one second before calling to see if the operation is complete. Double your wait time for each successive iteration of calls until the operation is complete.
`state`	AsyncRequestState (enumeration of type string)	Required. The `AsyncRequestState` object has one of four possible values. • `Queued`: This call hasn't started. It's waiting in a queue. • `InProgress`: This call has started but hasn't been completed. • `Completed`: This call has been completed. • `Error`: An error occurred. See the `statusCode` for more information.
`stateDetail`	string	Indicates which component is being deployed or which Apex test class is running. This field is available in API version 16.0 and later and is relevant only for the deploy() call.
`stateDetailLastModifiedDate`	dateTime	The date and time when the `stateDetail` field was last modified. This field is available in API version 16.0 and later and is relevant only for the deploy() call.
`statusCode`	StatusCode (enumeration of type string)	If an error occurred during the `create()`, `update()`, `delete()`, or `deploy()` call, a status code is returned, and the message that corresponds to the status code is returned in the `message` field. For a description of each StatusCode value, see StatusCode in the SOAP API Developer Guide.

CancelDeployResult

Contains information about a deployment cancellation—whether the cancellation completed and the deployment ID.

The asynchronous metadata call cancelDeploy() returns a `CancelDeployResult` object.

Version

Available in API version 30.0 and later.

`CancelDeployResult` has the following properties.

Name	Type	Description
`done`	boolean	Indicates whether the deployment cancellation, which is started through `cancelDeploy()`, has completed (`true`) or not (`false`). When a deployment hasn't started yet and is still in the queue, the deployment is canceled immediately with the `cancelDeploy()` call and this field returns `true`. Otherwise, this field returns `false` when the cancellation is in progress.
`id`	ID	ID of the deployment being canceled.

DeployResult

Contains information about the success or failure of the associated `deploy()` call.

The asynchronous metadata call checkDeployStatus() returns a DeployResult object.

In API version 29.0, Salesforce moved several properties from the AsyncResult on page 114 object to the DeployResult object to improve the process for getting information about deployments. For more information about these changes, see deploy() on page 56.

For API version 29.0 and later, the DeployResult object has these properties.

Name	Type	Description
`id`	ID	ID of the component being deployed.
`canceledBy`	ID	The ID of the user who canceled the deployment. This field is available in API version 30.0 and later.
`canceledByName`	string	The full name of the user who canceled the deployment. This field is available in API version 30.0 and later.
`checkOnly`	boolean	Indicates whether this deployment is used to check the validity of the deployed files without changing the organization (`true`) or not (`false`). A check-only deployment doesn't deploy any components or change the organization in any way.
`completedDate`	dateTime	Timestamp for when the deployment process ended.

Name	Type	Description
`createdBy`	ID	The ID of the user who created the deployment. This field is available in API version 30.0 and later.
`createdByName`	string	The full name of the user who created the deployment. This field is available in API version 30.0 and later.
`createdDate`	dateTime	Timestamp for when the `deploy()` call was received.
`details`	DeployDetails[]	Provides the details of a deployment that is in-progress or ended, if the `includeDetails` parameter is set to `true` in the checkDeployStatus() call.
`done`	boolean	Indicates whether the server finished processing the `deploy()` call for the specified `id`.
`errorMessage`	string	Message corresponding to the values in the `errorStatusCode` field, if any.
`errorStatusCode`	string	If an error occurred during the deploy() call, a status code is returned, and the message corresponding to the status code is returned in the `errorMessage` field. For a description of each StatusCode value, see StatusCode in the *SOAP API Developer Guide*.
`ignoreWarnings`	boolean	Optional. Defaults to `false`. Specifies whether a deployment continues even if the deployment generates warnings. Don't set this argument to `true` for deployments to production organizations.
`lastModifiedDate`	dateTime	Timestamp of the last update for the deployment process.
`numberComponentErrors`	int	The number of components that generated errors during this deployment.
`numberComponentsDeployed`	int	The number of components deployed in the deployment process. Use this value with the `numberComponentsTotal` value to get an estimate of the deployment's progress.
`numberComponentsTotal`	int	The total number of components in the deployment. Use this value with the `numberComponentsDeployed` value to get an estimate of the deployment's progress.
`numberTestErrors`	int	The number of Apex tests that have generated errors during this deployment.
`numberTestsCompleted`	int	The number of completed Apex tests for this deployment. Use this value with the `numberTestsTotal` value to get an estimate of the deployment's test progress.
`numberTestsTotal`	int	The total number of Apex tests for this deployment. Use this value with the `numberTestsCompleted` value to get an estimate of the deployment's test progress. The value in this field isn't accurate until the deployment has started running tests for the components being deployed.
`numFiles`	int	The total number of files included in this deployment. This field is available in API version 64.0 and later.

Name	Type	Description
`runTestsEnabled`	boolean	Indicates whether Apex tests were run as part of this deployment (`true`) or not (`false`). Tests are either automatically run as part of a deployment or can be set to run in `DeployOptions` for the deploy() call. For information on when tests are automatically run, see Running Tests in a Deployment. This field is available in API version 30.0 and later.
`rollbackOnError`	boolean	Optional. Defaults to true. Indicates whether any failure causes a complete rollback (`true`) or not (`false`). If `false`, whatever set of actions can be performed without errors are performed, and errors are returned for the remaining actions. This parameter must be set to `true` if you're deploying to a production organization.
`startDate`	dateTime	Timestamp for when the deployment process began.
`stateDetail`	string	Indicates which component is being deployed or which Apex test class is running.
`status`	DeployStatus (enumeration of type string)	Indicates the current state of the deployment. The valid values are: • `Pending` • `InProgress` • `Succeeded` • `SucceededPartial` • `Failed` • `Canceling` • `Canceled`
`success`	boolean	Indicates whether the deployment was successful (`true`) or not (`false`).
`zipSize`	int	The size of the deployment .zip file in bytes. This field is available in API version 64.0 and later.

DeployDetails

These fields provide more information for the `details` field of the DeployResult object, if the `includeDetails` parameter is set to (`true` in the deploy() call.

Note: While a deployment is still in-progress, the DeployDetails object only contains `componentFailures` data. After the deployment process finishes, the other fields populate with the data for the entire deployment.

Name	Type	Description
`componentFailures`	DeployMessage[]	One or more DeployMessage objects containing deployment errors for each component.
`componentSuccesses`	DeployMessage[]	One or more DeployMessage objects containing successful deployment details for each component.

Name	Type	Description
`retrieveResult`	RetrieveResult	If the `performRetrieve` parameter was specified for the deploy() call, a retrieve() call is performed immediately after the deploy() process completes. This field contains the results of that retrieval.
`runTestResult`	RunTestsResult	If tests were run for the deploy() call, this field contains the test results. While a deployment is still in-progress, this field only contains error data. After the deployment process finishes, this field populates with the data for the entire deployment.

For API version 28.0 and earlier, the DeployResult object has the following properties.

Name	Type	Description
`id`	ID	ID of the component being deployed.
`messages`	DeployMessage[]	Contains information about the success or failure of a deploy() call.
`retrieveResult`	RetrieveResult	If the `performRetrieve` parameter was specified for the deploy() call, a retrieve() call is performed immediately after the deploy() process completes. This field contains the results of that retrieval.
`runTestResult`	RunTestsResult	If tests were run for the deploy() call, this field contains the test results.
`success`	boolean	Indicates whether the deployment was successful (`true`) or not (`false`).

DeployMessage

Each DeployResult object contains one or more DeployMessage objects. Each DeployMessage object contains information about the deployment success or failure of a component in the deployment `.zip` file:

As of the Spring '20 release, only authenticated users can access DeployMessage objects.

Name	Type	Description
`changed`	boolean	If `true`, the component was changed as a result of this deployment. If `false`, the deployed component was the same as the corresponding component already in the organization.
`columnNumber`	int	A text file represents each component. If an error occurred during deployment, this field represents the column of the text file where the error occurred.
`componentType`	string	The metadata type of the component in this deployment. This field is available in API version 30.0 and later.
`created`	boolean	If `true`, the component was created as a result of this deployment. If `false`, the component was either deleted or modified as a result of the deployment.
`createdDate`	dateTime	The date and time when the component was created as a result of this deployment. This field is available in API version 30.0 and later.

Name	Type	Description
`deleted`	boolean	If `true`, the component was deleted as a result of this deployment. If `false`, the component was either new or modified as a result of the deployment.
`fileName`	string	The name of the file in the `.zip` file used to deploy this component.
`fullName`	string	The full name of the component. Inherited from Metadata, this field is defined in the WSDL for this metadata type. It must be specified when creating, updating, or deleting. See `createMetadata()` to see an example of this field specified for a call.
`id`	ID	ID of the component being deployed.
`lineNumber`	int	A text file represents each component. If an error occurred during deployment, this field represents the line number of the text file where the error occurred.
`problem`	string	If an error or warning occurred, this field contains a description of the problem that caused the compile to fail.
`problemType`	DeployProblemType (enumeration of type string)	Indicates the problem type. The problem details are tracked in the `problem` field. The valid values are: • `Warning` • `Error` This field is available in API version 18.0 and later. Before version 18.0, there was no distinction between warnings and errors. All problems were treated as errors and prevented a successful deployment.
`success`	boolean	Indicates whether the component was successfully deployed (`true`) or not (`false`).

RunTestsResult

A RunTestsResult object has the following properties

Name	Type	Description
`apexLogId`	string	The ID of an ApexLog object that is created at the end of a test run. The ApexLog object is created if there is an active trace flag on the user running an Apex test, or on a class or trigger being executed. This field is available in API version 35.0 and later.
`codeCoverage`	CodeCoverageResult[]	An array of one or more CodeCoverageResult objects that contains the details of the code coverage for the specified unit tests.
`codeCoverageWarnings`	CodeCoverageWarning[]	An array of one or more code coverage warnings for the test run. The results include both the total number of lines that could have been executed, as well as the number, line, and column positions of code that was not executed.

Name	Type	Description
failures	RunTestFailure[]	An array of one or more RunTestFailure objects that contain information about the unit test failures, if there are any.
flowCoverage	FlowCoverageResult[]	An array of results from test runs that executed flows. This field is available in API version 44.0 and later.
flowCoverageWarnings	FlowCoverageWarning[]	An array of warnings generated by test runs that executed flows. This field is available in API version 44.0 and later.
numFailures	int	The number of failures for the unit tests.
numTestsRun	int	The number of unit tests that were run.
successes	RunTestSuccess[]	An array of one or more RunTestSuccess objects that contain information about successes, if there are any.
totalTime	double	The total cumulative time spent running tests, in milliseconds. This can be helpful for performance monitoring.

CodeCoverageResult

The RunTestsResult object contains this object. It contains information about whether or not the compile of the specified Apex and run of the unit tests was successful.

Name	Type	Description
dmlInfo	CodeLocation[]	For each class or trigger tested, for each portion of code tested, this property contains the DML statement locations, the number of times the code was executed, and the total cumulative time spent in these calls. This can be helpful for performance monitoring.
id	ID	The ID of the CodeLocation. The ID is unique within an organization.
locationsNotCovered	CodeLocation[]	For each class or trigger tested, if any code is not covered, the line and column of the code not tested, and the number of times the code was executed.
methodInfo	CodeLocation[]	For each class or trigger tested, the method invocation locations, the number of times the code was executed, and the total cumulative time spent in these calls. This can be helpful for performance monitoring.
name	string	The name of the class or trigger covered.
namespace	string	The namespace that contained the unit tests, if one is specified.

Name	Type	Description
numLocations	int	The total number of code locations.
numLocationsNotCovered	int	The number of code locations not covered by any class or trigger.
soqlInfo	CodeLocation[]	For each class or trigger tested, the location of SOQL statements in the code, the number of times this code was executed, and the total cumulative time spent in these calls. This can be helpful for performance monitoring.
type	string	Do not use. In early, unsupported releases, used to specify class or package.

CodeCoverageWarning

The RunTestsResult object contains this object. It contains information about the Apex class which generated warnings.

This object has the following properties.

Name	Type	Description
id	ID	The ID of the CodeLocation. The ID is unique within an organization.
message	string	The message of the warning generated.
name	string	The namespace that contained the unit tests, if one is specified.
namespace	string	The namespace that contained the unit tests, if one is specified.

RunTestFailure

The RunTestsResult object returns information about failures during the unit test run.

This object has these properties.

Name	Type	Description
id	ID	The ID of the class which generated failures.
message	string	The failure message.
methodName	string	The name of the method that failed.
name	string	The name of the class that failed.
namespace	string	The namespace that contained the class, if one was specified.

Name	Type	Description
`seeAllData`	boolean	Indicates whether the test method has access to organization data (`true`) or not (`false`). This field is available in API version 33.0 and later.
`stackTrace`	string	The stack trace for the failure.
`time`	double	The time spent running tests for this failed operation, in milliseconds. This can be helpful for performance monitoring.
`type`	string	Do not use. In early, unsupported releases, used to specify class or package.

FlowCoverageResult

This object contains information about the flow version and the number of elements executed by the test run. This object is available in API version 44.0 and later.

Name	Type	Description
`elementsNotCovered`	string	List of elements in the flow version that weren't executed by the test run.
`flowId`	string	The ID of the flow version. The ID is unique within an org.
`flowName`	string	The name of the flow that was executed by the test run.
`flowNamespace`	string	The namespace that contains the flow, if one is specified.
`numElements`	int	The total number of elements in the flow version.
`numElementsNotCovered`	int	The number of elements in the flow version that weren't executed by the test run
`processType`	FlowProcessType (enumeration of type string)	The process type of the flow version.

FlowCoverageWarning

This object contains information about the flow version that generated warnings. This object is available in API version 44.0 and later.

Name	Type	Description
`flowId`	string	The ID of the flow version that generated the warning.
`flowName`	string	The name of the flow that generated the warning. If the warning applies to the overall test coverage of flows within your org, this value is null.
`flowNamespace`	string	The namespace that contains the flow, if one was specified.
`message`	string	The message of the warning that was generated.

RunTestSuccess

The RunTestsResult object returns information about successes during the unit test run.

This object has these properties.

Name	Type	Description
`id`	ID	The ID of the class which generated the success.
`methodName`	string	The name of the method that succeeded.
`name`	string	The name of the class that succeeded.
`namespace`	string	The namespace that contained the unit tests, if one is specified.
`seeAllData`	boolean	Indicates whether the test method has access to organization data (`true`) or not (`false`). This field is available in API version 33.0 and later.
`time`	double	The time spent running tests for this operation. This can be helpful for performance monitoring.

CodeLocation

The RunTestsResult object contains this object in a number of fields.

This object has these properties.

Name	Type	Description
`column`	int	The column location of the Apex tested.
`line`	int	The line location of the Apex tested.
`numExecutions`	int	The number of times the Apex was executed in the test run.
`time`	double	The total cumulative time spent at this location. This can be helpful for performance monitoring.

DescribeMetadataResult

Contains information about the organization that is useful for developers working with declarative metadata.

The `describeMetadata()` call returns a DescribeMetadataResult object.

Each DescribeMetadataResult object has the following properties:

Name	Type	Description
`metadataObjects`	DescribeMetadataObject[]	One or more metadata components and their attributes.
`organizationNamespace`	string	The namespace of the organization. Specify only for Developer Edition organizations that can contain a managed package. The managed package has a namespace specified when it is created.
`partialSaveAllowed`	boolean	Indicates whether `rollbackOnError` is allowed (`true`) or not (`false`). This value is always : • `false` in production organizations. • the opposite of `testRequired`.
`testRequired`	boolean	Indicates whether tests are required (`true`) or not (`false`). This value is always the opposite of `partialSaveAllowed`.

DescribeMetadataObject

This object is returned as part of the DescribeMetadataResult. Each DescribeMetadataObject has the following properties:

Name	Type	Description
`childXmlNames`	string[]	List of child sub-components for this component.
`directoryName`	string	The name of the directory in the `.zip` file that contains this component.
`inFolder`	boolean	Indicates whether the component is in a folder (`true`) or not (`false`). For example, documents, email templates and reports are stored in folders.
`metaFile`	boolean	Indicates whether the component requires an accompanying metadata file. For example, documents, classes, and s-controls are components that require an additional metadata file.
`suffix`	string	The file suffix for this component.
`xmlName`	string	The name of the root element in the metadata file for this component. This name also appears in the `Packages` > `types` > `name` field in the manifest file `package.xml`.

DescribeValueTypeResult

Contains information about a value type that's useful for developers working with declarative metadata.

The `describeValueType()` call returns a DescribeValueTypeResult object.

Each DescribeValueTypeResult object has these properties.

Name	Type	Description
`apiCreatable`	boolean	Indicates whether this value type can be created through the createMetadata() call (`true`) or not (`false`). This field is available in API version 36.0 and later.
`apiDeletable`	boolean	Indicates whether this value type can be created through the deleteMetadata() call (`true`) or not (`false`). This field is available in API version 36.0 and later.
`apiReadable`	boolean	Indicates whether this value type can be created through the readMetadata() call (`true`) or not (`false`). This field is available in API version 36.0 and later.
`apiUpdatable`	boolean	Indicates whether this value type can be created through the updateMetadata() call (`true`) or not (`false`). This field is available in API version 36.0 and later.
`parentField`	ValueTypeField	Information about the parent of this value type. Parent field information is useful for metadata types that are specified with the parent in their name, such as custom fields, email templates, workflow rules, and reports. For example, the full name of a custom field includes the sObject that contains it (for example, `Account.field1__c`). Similarly, the full name of an email template includes the folder where the template is stored (for example, `MyFolder/EmailTemplate1`). If the value type has no parent, this field is null. This field is available in API version 36.0 and later.
`valueTypeFields`	ValueTypeField[]	One or more metadata components and their attributes.

ValueTypeField

This object is returned as part of the DescribeValueTypeResult and represents the metadata for one field. Each ValueTypeField has these properties.

Name	Type	Description
`fields`	ValueTypeField	The `ValueTypeField` object for the next field, if any.
`foreignKeyDomain`	string	If `isForeignKey` is `True`, `foreignKeyDomain` is the type of object, such as Account or Opportunity.
`isForeignKey`	boolean	`True` if the field is a foreign key. That means this field is the primary key in a different database table.
`isNameField`	boolean	`True` if this value type field is a `fullName` field; otherwise `False`.
`minOccurs`	int	1 if this field is required, 0 otherwise.

Name	Type	Description
`name`	string	The name of this value type field. The name is null for parent fields.
`picklistValues`	PicklistEntry[]	The individual picklist values if the field is a picklist.
`soapType`	string	The data type of the field, such as boolean or double.
`valueRequired`	boolean	Required. Indicates whether this value type field must have a value (`true`) or can be null (`false`).

PicklistEntry

This object is returned as part of the DescribeValueTypeResult and represents the metadata for one picklist value. Each PicklistEntry has these properties.

Name	Type	Description
`active`	boolean	`True` if the picklist value is active and displayed in the drop-down list for the picklist field in the user interface. `False` if the picklist value is inactive.
`defaultValue`	boolean	`True` if the picklist value is a default value, otherwise, `False`.
`label`	string	Display name of this item in the picklist.
`validFor`	string	
`value`	string	Value of this item in the picklist.

ReadResult

Contains result information for the `readMetadata` call.

Version

Available in API version 30.0 and later.

Properties

Name	Type	Description
`records`	Metadata[]	An array of metadata components returned from `readMetadata()`.

RetrieveResult

The `retrieve()` call returns an array of RetrieveResult objects.

A RetrieveResult object has these fields.

Name	Type	Description
`done`	boolean	Required. Indicates whether the retrieve() on page 75 call is completed (`true`) or not (`false`). This field is available in API version 31.0 and later.
`errorMessage`	string	If an error occurs during the retrieve() on page 75 call, this field contains a descriptive message about this error. This field is available in API version 31.0 and later.
`errorStatusCode`	StatusCode	If an error occurs during the retrieve() on page 75 call, this field contains the status code for this error. This field is available in API version 31.0 and later. For a description of each StatusCode value, see StatusCode in the SOAP API Developer Guide.
`fileProperties`	FileProperties[]	Contains information about the properties of each component in the `.zip` file, and the manifest file `package.xml`. One object per component is returned.
`id`	ID	ID of the component being retrieved.
`messages`	RetrieveMessage[]	Contains information about the success or failure of the retrieve() on page 75 call.
`status`	RetrieveStatus (enumeration of type string)	The status of the retrieve() on page 75 call. Valid values are: • `Pending` • `InProgress` • `Succeeded` • `Failed` This field is available in API version 31.0 and later.
`success`	boolean	Indicates whether the retrieve() on page 75 call was successful (`true`) or not (`false`). This field is available in API version 31.0 and later.
`zipFile`	base64Binary	The zip file returned by the retrieve request. Base 64-encoded binary data. Before making an API call, client applications must encode the binary attachment data as base64. Upon receiving a response, client applications must decode the base64 data to binary. This conversion is handled for you by a SOAP client.

FileProperties

This component contains information about the properties of each component in the `.zip` file, and the manifest file `package.xml`. One object per component is returned. This component doesn't contain information about any associated metadata files in the `.zip` file, only the component files and manifest file. FileProperties contains the following properties:

Name	Type	Description
`createdById`	string	Required. ID of the user who created the file.
`createdByName`	string	Required. Name of the user who created the file.
`createdDate`	dateTime	Required. Date and time when the file was created.

Name	Type	Description
`fileName`	string	Required. Name of the file.
`fullName`	string	Required. The file developer name used as a unique identifier for API access. The value is based on the `fileName` but the characters allowed are more restrictive. The `fullName` can contain only underscores and alphanumeric characters. It must be unique, begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
`id`	string	Required. ID of the file.
`lastModifiedById`	string	Required. ID of the user who last modified the file.
`lastModifiedByName`	string	Required. Name of the user who last modified the file.
`lastModifiedDate`	dateTime	Required. Date and time that the file was last modified.
`manageableState`	ManageableState (enumeration of type string)	Indicates the manageable state of the specified component if it's contained in a package: • `beta` • `deleted` • `deprecated` • `deprecatedEditable` • `installed` • `installedEditable` • `released` • `unmanaged`
`namespacePrefix`	string	If any, the namespace prefix of the component.
`type`	string	Required. The metadata type, such as `CustomObject`, `CustomField`, or `ApexClass`.

RetrieveMessage

RetrieveResult on page 129 returns this object, which contains information about the success or failure of the retrieve() on page 75 call. One object per problem is returned:

Name	Type	Description
`fileName`	string	The name of the file in the retrieved `.zip` file where a problem occurred.
`problem`	string	A description of the problem that occurred.

SEE ALSO:

retrieve()

SaveResult

Contains result information for the `createMetadata`, `updateMetadata`, or `renameMetadata` call.

Version

Available in API version 30.0 and later.

Properties

Name	Type	Description
`errors`	Error[]	An array of errors returned if the operation wasn't successful.
`fullName`	string	The full name of the component processed.
`success`	boolean	Indicates whether the operation was successful (`true`) or not (`false`).

DeleteResult

Contains result information for the `deleteMetadata` call.

Version

Available in API version 30.0 and later.

Properties

Name	Type	Description
`errors`	Error[]	An array of errors returned if the operation wasn't successful.
`fullName`	string	The full name of the deleted component.
`success`	boolean	Indicates whether the deletion was successful (`true`) or not (`false`).

UpsertResult

Contains information about the result of the associated `upsertMetadata()` call.

Version

Available in API version 31.0 and later.

Properties

Name	Type	Description
`created`	boolean	Indicates whether the upsert operation resulted in the creation of the component (`true`) or not (`false`). If `false` and the upsert operation was successful, the component was updated.
`errors`	Error[]	An array of errors that were returned if the operation wasn't successful.
`fullName`	string	The full name of the component that was created or updated if the operation was successful.
`success`	boolean	Indicates whether the operation was successful (`true`) or not (`false`).

Error

Represents an error that occurred during a synchronous CRUD (`createMetadata()`, `updateMetadata()`, or `deleteMetadata()`) operation.

Version

Available in API version 30.0 and later.

Properties

Name	Type	Description
`extendedErrorDetails`	ExtendedErrorDetails	More details about the error, including an extended error code and extra error properties, when available. Reserved for future use. For a description of the ExtendedErrorDetails element, see ExtendedErrorDetails in the *SOAP API Developer Guide*.
`fields`	string[]	An array containing names of fields that affected the error condition.
`message`	string	The error message text.
`statusCode`	StatusCode	A status code corresponding to the error. For a description of each StatusCode value, see StatusCode in the *SOAP API Developer Guide*.

CHAPTER 12 Metadata Types

Metadata API enables you to access some entities and feature settings that you can customize in the user interface.

Note:

- Metadata type names are case-sensitive. Specifying a type name with an invalid case results in a deployment error.
- Metadata types don't always correspond directly to their related data types. In some cases, the information is accessible but not organized as expected. For example, dependent picklists are exposed as a type of picklist, not a separate metadata type.
- The wildcard character doesn't apply to metadata types for feature settings, like AccountSettings. The wildcard applies only when retrieving all settings and not an individual setting. See Settings.

Metadata Coverage Report

Launch the Metadata Coverage report to determine supported metadata components. The Metadata Coverage report is the ultimate source of truth for metadata coverage across several channels. These channels include Metadata API, scratch org source tracking, unlocked packages, second-generation managed packages, classic managed packages, and more.

Unsupported Metadata Types

Some Salesforce features have metadata types that aren't available in Metadata API. These metadata types can't be retrieved or deployed with Metadata API. To make changes to these types, you must do it manually in each of your organizations.

Special Behavior in Metadata API Deployments

Important considerations for specific types and contents of a deployment.

Data Cloud Metadata Types

Check out the metadata types that are used for development in Data Cloud.

AccountRelationshipShareRule

The rule that determines which object records are shared, how they're shared, the account relationship type that shares the records, and the level of access granted to the records.

AccountingFieldMapping

Represents the accounting field mappings to organize your data and bring it to ledger entry records.

AccountingModelConfig

Represents the mapping of the financial data model to a logical data model and configuration for the generation of Transaction Journal records.

ActionLinkGroupTemplate

Represents the action link group template. Action link templates let you reuse action link definitions and package and distribute action links. An action link is a button on a feed element. Clicking on an action link can take a user to another Web page, initiate a file download, or invoke an API call to an external server or Salesforce. Use action links to integrate Salesforce and third-party services into the feed. Every action link belongs to an action link group and action links within the group are mutually exclusive.

ActionPlanTemplate

Represents the instance of an action plan template.This type extends the Metadata metadata type and inherits its `fullName` field.

ActionableListDefinition

Represents the data source definition details associated with an actionable list.

AdvAccountForecastSet

Represents the forecast sets that define the forecast configurations for each business unit or different groups of accounts. With separate forecast sets at account or business unit level, you can focus on account-specific data and manage configuration updates for one business unit without impacting any other business unit's data.

AffinityScoreDefinition

Represents the affinity information used in calculations to analyze and categorize contacts for marketing purposes.

AIApplication

Represents an instance of an AI application. For example, Einstein Prediction Builder. This type extends the Metadata metadata type and inherits its `fullName` field.

AIApplicationConfig

Additional prediction information related to an AI application. This type extends the Metadata metadata type and inherits its `fullName` field.

AIScoringModelDefinition

Represents information about a machine learning model that's used by the Scoring Framework for Industries Cloud Einstein. The machine learning model is used for scoring, including its configuration.

AIUsecaseDefinition

Represents a collection of fields in your Salesforce org used to define a machine learning use case and get real-time predictions.

AnalyticsDashboard

Represents a Tableau Next dashboard.

AnalyticSnapshot

Represents a reporting snapshot. A reporting snapshot lets you report on historical data. Authorized users can save tabular or summary report results to fields on a custom object, then map those fields to corresponding fields on a target object. They can then schedule when to run the report to load the custom object's fields with the report's data. Reporting snapshots enable you to work with report data similarly to how you work with other records in Salesforce.

AnalyticsVisualization

Represents a Tableau Next visualization.

AnalyticsWorkspace

Represents a Tableau Next workspace.

AnimationRule

Represents criteria for determining when an animation is displayed to Path users.This type extends the Metadata metadata type and inherits its `fullName` field.

AppFrameworkTemplateBundle

Represents the app framework template bundle. Use these templates for Data Cloud and Tableau Next assets.

ArticleType

Represents the metadata associated with an article type.

ApexClass

Represents an Apex class. An Apex class is a template or blueprint from which Apex objects are created. Classes consist of other classes, user-defined methods, variables, exception types, and static initialization code.

ApexComponent

Represents a Visualforce component.

ApexEmailNotifications

The ApexEmailNotifications type allows you to define users and email addresses that receive email for unhandled Apex errors. Flow errors can also use this metadata type.

ApexPage

Represents a Visualforce page.

ApexTestSuite

Represents a suite of Apex test classes to include in a test run.

ApexTrigger

Represents an Apex trigger. A trigger is Apex code that executes before or after specific data manipulation language (DML) events occur, such as before object records are inserted into the database, or after records have been deleted.

AppMenu

Represents the app menu or the Salesforce mobile navigation menu. Reserved for future use.

AppointmentAssignmentPolicy

Represents the information about a resource assignment rule. This type extends the Metadata metadata type and inherits its `fullName` field.

AppointmentSchedulingPolicy

Represents a set of rules for scheduling appointments using Lightning Scheduler. This type extends the Metadata metadata type and inherits its `fullName` field.

ApprovalProcess

Represents the metadata associated with an approval process. An approval process automates how records are approved in Salesforce. An approval process specifies each step of approval, including who to request approval from and what to do at each point of the process.

AssignmentRules

Represents assignment rules that allow you to automatically route cases to the appropriate users or queues. You can access rules metadata for all applicable objects, for a specific object, or for a specific rule on a specific object.

AssessmentQuestion

Represents the container object that stores the questions required for an assessment.

AssessmentQuestionSet

Represents the container object for Assessment Questions.

Audience

Represents the audience in an Experience Builder site. An audience consists of different types of criteria, where the audience can be assigned and used for targeting in a site. This type extends the Metadata metadata type and inherits its `fullName` field.

AuraDefinitionBundle

Represents an Aura definition bundle. A bundle contains an Aura definition, such as an Aura component, and its related resources, such as a JavaScript controller. The definition can be a component, application, event, interface, or a tokens collection.

AuthProvider

Represents an authentication provider (auth provider). An auth provider lets users log in to Salesforce from an external service provider such as Facebook, Google, or GitHub. This type extends the Metadata metadata type and inherits its `fullName` field.

AutoResponseRules

Represents an auto-response rule that sets conditions for sending automatic email responses to lead or case submissions based on the attributes of the submitted record. You can access rules metadata for all applicable objects, for a specific object, or for a specific rule on a specific object.

BatchCalcJobDefinition

Represents a Data Processing Engine definition.

BatchProcessJobDefinition

Represents the details of a Batch Management job definition. This type extends the Metadata metadata type and inherits its `fullName` field.

BillingSettings

Represents the settings for Salesforce Billing.

BlacklistedConsumer

Represents a connected app that is inaccessible to your Salesforce org's users. This type extends the Metadata metadata type and inherits its `fullName` field.

Bot

Represents a definition of an Einstein Bot configuration that can have one or more versions. Only one version can be active.

BotBlock

Represents the configuration details for a specific Einstein Bot block, including dialogs and variables.

BotTemplate

Represents the configuration details for a specific Einstein Bot template, including dialogs and variables.

BotVersion

Represents the configuration details for a specific Einstein Bot version, including dialogs and variables.

BrandingSet

Represents the definition of a set of branding properties for an Experience Builder site or for your org's Lightning Experience theme.

BriefcaseDefinition

Represents a briefcase definition. A briefcase makes selected records available for specific users and groups to view when they're offline in the Salesforce Field Service mobile app for iOS and Android. This type extends the Metadata metadata type and inherits its `fullName` field.

BusinessProcessGroup

Represents the surveys used to track customers' experiences across different stages in their lifecycle. This type extends the Metadata metadata type and inherits its `fullName` field.

CallCenter

Represents the Call Center definition used to integrate Salesforce with a third-party computer-telephony integration (CTI) system, a partner telephony system, or partner Contact Center as a Service (CCaaS) system.

CallCenterRoutingMap

Represents the mapping between a user or queue in a Salesforce org to a user or queue in an external system's call center.

CallCoachingMediaProvider

Represents the CallCoachingMediaProvider configuration. Use CallCoachingMediaProvider to configure which providers of voice recordings that Einstein Conversation Insights can use. For example, Sales Dialer can provide voice recordings. Einstein Conversation Insights then stores and analyzes call recordings to surface insights and trends in customer conversations.This type extends the Metadata metadata type and inherits its `fullName` field.

CampaignInfluenceModel

Represents a campaign influence model used by Customizable Campaign Influence. You can't configure Customizable Campaign Influence via the Metadata API, but you can add a campaign influence model.

CaseSubjectParticle

Represents the Social Business Rules custom format for the **Case Subject** field on cases created from inbound social posts.

CareBenefitVerifySettings

Represents the configuration settings for benefit verification requests.

CareLimitType

Defines the characteristics of limits on benefit provision.

CareSystemFieldMapping

Represents a mapping from source system fields to Salesforce objects and fields. This type extends the Metadata metadata type and inherits its `fullName` field.

CareProviderSearchConfig

Represents the information about the fields that appear in care provider search results.This type extends the Metadata metadata type and inherits its `fullName` field.

CareRequestConfiguration

Represents the details for a record type such as service request, drug request, or admission request. One or more record types can be associated with a care request.

Certificate

Represents a certificate used for digital signatures that verify that requests are coming from your org. Certificates are used for either authenticated single sign-on with an external website, or when using your org as an identity provider. This type extends the Metadata With Content metadata type and inherits its `content` and `fullName` fields.

ChatterExtension

Represents the metadata used to describe a Rich Publisher App that's integrated with the Chatter publisher.

ChoiceList

Represents the `Choicelist` dropdown field that's used for pre-chat.

ClaimFinancialSettings

Represents the configuration settings for Insurance Claim Financial Services.

ClauseCatgConfiguration

Represents the configuration about the clause category that can be used to categorize your disclosure and compliance reports from standardized disclosure templates in a response document.

CleanDataService

Represents a data service that adds and updates data in standard objects.

CMSConnectSource

Represents the connection information for external content management systems that feed content to Experience Builder sites. This type extends the Metadata metadata type and inherits its `fullName` field.

Community (Zone)

Represents a zone that contains Ideas or Chatter Answers objects. Zones are shared by the Ideas, Answers, and Chatter Answers features, allowing you to view and create zones from those locations.This type extends the Metadata metadata type and inherits its `fullName` field.

CommerceSettings

Represents settings for various Commerce features.

CommunityTemplateDefinition

Represents the definition of an Experience Builder site template. This type extends the Metadata metadata type and inherits its `fullName` field.

CommunityThemeDefinition

Represents the definition of a theme for an Experience Builder site. This type extends the Metadata metadata type and inherits its `fullName` field.

ConnectedApp

Represents a connected app configuration. A connected app enables an external application to integrate with Salesforce using APIs and standard protocols, such as SAML, OAuth, and OpenID Connect. Connected apps use these protocols to authenticate, authorize, and provide single sign-on (SSO) for external apps. The external apps that are integrated with Salesforce can run on the customer success platform, other platforms, devices, or SaaS subscriptions.

ContentAsset

Represents the metadata for creating an asset file. Asset files enable a Salesforce file to be used for org setup and configuration purposes. This type extends the MetadataWithContent metadata type and inherits its `content` and `fullName` fields.

ContentTypeBundle

Represents the definition of enhanced custom content types for use with enhanced CMS workspaces. When you create an enhanced custom content type, deploy this bundle to your org. Enhanced custom content types are displayed as forms with defined fields. When deployed, enhanced custom content types are available for use with enhanced LWR site channels. To use enhanced custom content types with Aura and non-enhanced LWR site channels, use enhanced CMS workspaces resources.

ContextDefinition

Represents the details of a context definition that describe the relationship between the node structures within a context.

ConversationMessageDefinition

Represents a messaging component in an Enhanced Messaging channel or Messaging for In-App and Web session.

ConversationVendorInfo

Represents the connection between the partner vendor system and the Service Cloud feature. For example, for Service Cloud Voice, this type contains information about the partner telephony system or Contact Center as a Service (CCaaS) system. For Bring Your Own Channel for Messaging or Bring Your Own Channel for CCaaS, this type contains information about the partner messaging system or CCaaS system.

ConvIntelligenceSignalRule

Represents the conversation intelligence signal rule. The rule triggers actions based on real-time intelligence signals from your telephony system or keywords mentioned by support reps or customers. The rule contains a set of conditions (subrules) and the filter logic used to evaluate those conditions to determine whether to trigger actions.

CorsWhitelistOrigin

Represents an origin in the CORS allowlist.

CspTrustedSite

Represents a trusted URL. For each CspTrustedSite component, you can specify Content Security Policy (CSP) directives and permissions policy directives. Each CSP directive allows Lightning components, third-party APIs, and WebSocket connections to access a resource type from the trusted URL. If the Permissions-Policy HTTP header is enabled, each permissions policy directive grants the trusted URL access to a browser feature. In API version 58.0 and earlier, CspTrustedSite components included only CSP directives and were referred to as CSP Trusted Sites.

CustomApplication

CustomApplication represents a custom or standard application. In API version 29.0 and earlier, CustomApplication represents only a custom application. An application is a list of tab references, with a description and a logo. This type extends the Metadata metadata type and inherits its `fullName` field.

CustomApplicationComponent

Represents a custom console component (Visualforce page) assigned to a CustomApplication that is marked as a Salesforce console. Custom console components extend the capabilities of Salesforce console apps. See Customize a Console with Custom Components in Salesforce Classic in Salesforce Help.

CustomFeedFilter

Represents a custom feed filter that limits the feed view to feeds from the Cases object. The custom feed filter shows only feed items that satisfy the criteria specified in the CustomFeedFilter definition. This type extends the Metadata metadata type and inherits its `fullName` field.

CustomFieldDisplay

Represents the view type assigned to product attribute custom fields. This type extends the Metadata metadata type and inherits its `fullName` field.

CustomHelpMenuSection

Represents the section of the Lightning Experience help menu that the admin added to display custom, org-specific help resources for the org. The custom section contains help resources added by the admin. This type extends the Metadata metadata type and inherits its `fullName` field.

CustomIndex

Represents an index used to increase the speed of queries.This type extends the Metadata metadata type and inherits its `fullName` field.

CustomLabels

The CustomLabels metadata type allows you to create custom labels that can be localized for use in different languages, countries, and currencies.

Custom Metadata Types (CustomObject)

Represents the metadata associated with a custom metadata type.

CustomNotificationType

Represents the metadata associated with a custom notification type.

CustomObject

Represents a custom object that stores data unique to your org or an external object that maps to data stored outside your org.

CustomObjectTranslation

This metadata type allows you to translate custom objects for a variety of languages.

CustomPageWebLink

Represents a custom link defined in a home page component.

CustomPermission

Represents a permission that grants access to a custom feature. This type extends the Metadata metadata type and inherits its `fullName` field.

CustomSite

Represents a Salesforce site. Create public websites and applications that are directly integrated with your Salesforce organization, but don't require users to log in with a username and password.

CustomTab

Represents a custom tab. Custom tabs let you display custom object data or other web content in Salesforce. When you add a custom tab to an app in Salesforce Classic, it appears as a tab. When you add a custom tab to an app in Lightning Experience, it appears as an item in the app's navigation bar and in the App Launcher. When a tab displays a custom object, the tab name is the same as the custom object name. For page, s-control, or URL tabs, the name is arbitrary.

CustomValue

Represents the definition of a value used in a global value set or local custom picklist. Custom picklist fields can be local and unique, or can inherit their values from a global picklist (called a *global value set* in API version 38.0). This type extends the Metadata metadata type and inherits its `fullName` field.

Dashboard

Represents a dashboard. Dashboards are visual representations of data that allow you to see key metrics and performance at a glance.

DataCategoryGroup

Represents a data category group.

DataObjectSearchIndexConf

Represents the source Data Cloud data model object (DMO) for Search Answers and holds the search index that Search Answers uses when searching DMO records.

DataWeaveResource

Represents the `DataWeaveScriptResource` class that is generated for all DataWeave scripts. DataWeave scripts can be directly invoked from Apex.

DecisionTable

Represents the information about a decision table.

DecisionTableDatasetLink

Represents the information about a dataset link associated with a decision table. In a dataset link, select an object for whose records, the decision table must provide an outcome.This type extends the Metadata metadata type and inherits its `fullName` field.

DecisionMatrixDefinition

Represents a definition of a decision matrix.

DelegateGroup

Represents a group of users who have the same administrative privileges. These groups are different from public groups used for sharing.

DigitalExperienceBundle

Represents a text-based code structure of your organization's workspaces, organized by workspace type, and each workspace's content items.

DigitalExperienceConfig

Represents details for your organization's workspaces, such as the site label, site URL path prefix, and workspace type.

DisclosureDefinition

Represents information that defines a disclosure type, such as details of the publisher or vendor who created or implemented the report.

DisclosureDefinitionVersion

Represents the version information about the disclosure definition.

DisclosureType

Represents the types of disclosures that are done by an individual or an organization and the associated metadata.

DiscoveryAIModel

Represents the metadata associated with a model used in Einstein Discovery.

DiscoveryGoal

Represents the metadata associated with an Einstein Discovery prediction definition.

DiscoveryStory

Represents the metadata associated with a story used in Einstein Discovery.

Document

Represents a Document. All documents must be in a document folder, such as `sampleFolder/TestDocument`.

DocumentCategory

Represents a document category.

DocumentCategoryDocumentType

Represents the junction between a DocumentCategory and a DocumentType. Puts a DocumentType in a DocumentCategory.

DocumentChecklistSettings

Represents an org's DocumentChecklistItem settings. This type extends the Metadata metadata type and inherits its `fullName` field.

DocumentType

Represents a document type.

DuplicateRule

Represents a rule that specifies how duplicate records in an object are detected. This type extends the Metadata metadata type and inherits its `fullName` field.

EclairGeoData

Represents an Analytics custom map chart. Custom maps are user-defined maps that are uploaded to Analytics and are used just as standard maps are. Custom maps are accessed in Analytics from the list of maps available with the map chart type.

EmailServicesFunction

Represents an email service. This type extends the Metadata metadata type and inherits its `fullName` field.

EmailTemplate

Represents a template for an email, mass email, list email, or Sales Engagement email. Supported in first-generation managed packages only.

EmbeddedServiceBranding

Represents the branding for each Embedded Service deployment. This type extends the Metadata metadata type and inherits its `fullName` field.

EmbeddedServiceConfig

Represents a setup node for creating an Embedded Service for Web deployment. This type extends the Metadata metadata type and inherits its `fullName` field.

EmbeddedServiceFieldService

Represents a setup node for creating an embedded Appointment Management deployment. This type extends the Metadata metadata type and inherits its `fullName` field.

EmbeddedServiceFlowConfig

Represents a setup node for creating an embedded flow. This type extends the Metadata metadata type and inherits its `fullName` field.

EmbeddedServiceLiveAgent

Represents a setup node for creating an embedded chat deployment. This type extends the Metadata metadata type and inherits its `fullName` field.

EmbeddedServiceMenuSettings

Represents a setup node for creating a channel menu deployment. Channel menus list the ways in which customers can contact your business. This type extends the Metadata metadata type and inherits its `fullName` field.

EnablementMeasureDefinition

Represents an Enablement measure, which specifies the job-related activity that a user performs to complete a milestone or outcome in an Enablement program. A measure identifies a source object and optional related objects, with optional field filters and filter logic, for tracking the activity. To avoid deployment errors, deploy measures before you deploy programs.

EnablementProgramDefinition

Represents an Enablement program, which includes exercises and measurable milestones to help users such as sales reps achieve specific outcomes related to your company's revenue goals.

EnblProgramTaskSubCategory

Represents a custom exercise type that an Enablement admin adds to an Enablement program in Program Builder. A custom exercise type also requires a corresponding EnblProgramTaskDefinition record for Program Builder and corresponding LearningItem and LearningItemType records for when users take the exercise in the Guidance Center.

EntitlementProcess

Represents the settings for an entitlement process.

EntitlementTemplate

Represents an entitlement template. Entitlement templates are predefined terms of customer support that you can quickly add to products. For example, you can create entitlement templates for Web or phone support so that users can easily add entitlements to products offered to customers.

EscalationRules

Represents case escalation rules to escalate cases automatically if they aren't resolved within a certain time. You can access rules metadata for all applicable objects, for a specific object, or for a specific rule on a specific object.

EventDelivery

Represents how an event instance maps to a target payload. Removed in API version 46.0. This type extends the Metadata metadata type and inherits its `fullName` field.

EventRelayConfig

Represents the configuration of an event relay, which relays platform events and change data capture events from Salesforce to Amazon EventBridge.

EventSubscription

Represents a subscription to an event type. Removed in API version 46.0. This type extends the Metadata metadata type and inherits its `fullName` field.

ExperienceBundle

Represents a text-based code structure of the settings and site components, such as pages, branding sets, and themes that make up an Experience Builder site. Developers can quickly update and deploy Experience Builder sites *programmatically* using their preferred development tools. This type extends the Metadata metadata type and inherits its `fullName` field.

ExperiencePropertyTypeBundle (Beta)

Represents a property type. When you create a custom property type for a Lightning web component, deploy this bundle to your org.

ExplainabilityMsgTemplate

Represents information about the template that contains the decision explanation message for a specified expression set step type.

ExpressionSetDefinition

Represents an expression set definition.

ExpressionSetMessageToken

Represents an interface to retrieve, deploy, create, update, or delete information on Expression Set Message Token.

ExpressionSetObjectAlias

Represents information about the alias of the source object that's used in an expression set.

ExternalAuthIdentityProvider

Represents an external authentication (auth) identity provider. An external auth identity provider links to an external credential and obtains OAuth tokens for outbound callouts to external systems.

ExternalClientApplication

Represents the header file for an external client application configuration.

ExternalCredential

Represents the details of how Salesforce authenticates to the external system.

ExternalAIModel

Represents the state of a given model for an Einstein for Service feature, such as Einstein Reply Recommendations.

ExternalServiceRegistration

Represents the External Service configuration for an org. This type extends the Metadata metadata type and inherits its `fullName` field.

ExtlClntAppConfigurablePolicies

Represents the policies for an external client app to disable or enable plugins.

ExtlClntAppGlobalOauthSettings

Represents the global settings for the OAuth plugin in an external client app. These settings include private and sensitive OAuth consumer information that can't be packaged and must not be added to source control.

ExtlClntAppMobileConfigurablePolicies (Beta)

Represents an external client app's mobile policies configuration.

ExtlClntAppMobileSettings (Beta)

Represents an external client app's mobile app settings, such as screen lock on a mobile device.

ExtlClntAppNotificationSettings (Beta)

Represents an external client app's notification subscriptions for mobile.

ExtlClntAppOauthConfigurablePolicies

Represents the policies configured by the admin for an OAuth-enabled external client app.

ExtlClntAppOauthSettings

Represents the settings configuration for the external client app's OAuth plugin.

ExtlClntAppPushConfigurablePolicies (Beta)

Represents an external client app's push notification policies configuration.

ExtlClntAppPushSettings (Beta)

Represents an external client app's push notification settings.

ExtlClntAppSamlConfigurablePolicies

Represents SAML configuration policies for an external client app. Use this type to configure Salesforce as an identity provider for SAML single sign-on (SSO). In this type of SSO configuration, users log in to a third-party service provider, such as Google, using their Salesforce credentials.

FeatureParameterBoolean

Represents a boolean feature parameter in the Feature Management App (FMA). Feature parameters let you drive app behavior and track activation metrics in subscriber orgs that install your package. This type extends the Metadata metadata type and inherits its `fullName` field.

FeatureParameterDate

Represents a date feature parameter in the Feature Management App (FMA). Feature parameters let you drive app behavior and track activation metrics in subscriber orgs that install your package. This type extends the Metadata metadata type and inherits its `fullName` field.

FeatureParameterInteger

Represents an integer feature parameter in the Feature Management App (FMA). Feature parameters let you drive app behavior and track activation metrics in subscriber orgs that install your package. This type extends the Metadata metadata type and inherits its `fullName` field.

FieldRestrictionRule

Represents a field visibility rule that controls whether a field is visible to a user, based on the field's inclusion in a field set. If Enhanced Personal Information Management setting was enabled before Spring '22, field visibility is based on the field's compliance categorization. This type extends the Metadata metadata type and inherits its `fullName` field.

FlexiPage

Represents the metadata associated with a Lightning page. A Lightning page represents a customizable screen made up of regions containing Lightning components.

Flow

Represents the metadata associated with a flow that encompasses the flow's structure, logic, and run-time behavior. It allows you to build dynamic applications that guide users through interactive screens, automate processes, and connect with various Salesforce and external services. This includes managing data operations like creating, updating, or deleting records, handling complex decisions, looping through collections, and invoking actions like Apex or external services to extend functionality. A flow contains options for API versioning, various execution environments, and detailed configuration of elements to design powerful automation solutions.

FlowCategory

Represents a list of flows that are grouped by category. Flows aren't added directly to a Lightning Bolt Solution. Instead, add the category the flows are in to the Lightning Bolt Solution. This type extends the Metadata metadata type and inherits its `fullName` field.

FlowDefinition

Represents the flow definition's description and active flow version number.

FlowTest

Represents the metadata associated with a flow test. Before you activate a record-triggered flow, you can test it to verify its expected results and identify flow run-time failures.

Folder

Represents a folder. This type extends the Metadata metadata type and inherits its `fullName` field.

ForecastingFilter

Represents the custom filter for including or excluding data from opportunity forecasts.

ForecastingFilterCondition

Represents the custom filter condition logic for including or excluding data from opportunity forecasts.

ForecastingSourceDefinition

Represents the object, measure, date type, and hierarchy that a forecast uses to project sales.

ForecastingType

Represents a forecast type.

ForecastingTypeSource

Represents the mapping of a forecasting source definition to a forecast type.

FuelType

Represents a custom fuel type in an org.

FuelTypeSustnUom

Represents a mapping between the custom fuel types and their corresponding unit of measure (UOM) values defined by a customer in an org.

FunctionReference

Represents information about a deployed Salesforce Function that can be invoked from the org. This type extends the Metadata metadata type and inherits its `fullName` field.

FundraisingConfig

Represents a collection of settings to configure the fundraising product.

GatewayProviderPaymentMethodType

Represents an entity that allows integrators and payment providers to choose an active payment to receive an order's payment data rather than allowing the Salesforce Order Management platform to select a default payment method. This object is available in API version 51 and later.

GenAiFunction

Represents an agent action that can be added to an AI agent.

GenAiPlanner

Represents a planner for an agent. It's a container for all the topics and actions used to interact with a large language model (LLM).

GenAiPlannerBundle

Represents a planner for an agent or agent template. It's a container for all the topics and actions used to interact with a large language model (LLM).

GenAiPlugin

Represents an agent topic, which is a category of actions related to a particular job to be done by AI agents.

GenAiPluginInstructionDef

Represents a topic instruction.

GenAiPromptTemplate

Represents the definition of a prompt template, including its related objects and fields.

GenAiPromptTemplateActv

Represents the activation status of a Salesforce-provided prompt template.

GlobalPicklist

Represents a global picklist, or the set of shared picklist values that custom picklist fields can use. In contrast, the custom picklist fields that are based on a global picklist are of type CustomValue. This type extends the Metadata metadata type and inherits its `fullName` field.

GlobalPicklistValue

Represents the definition of a value used in a global picklist. Custom picklist fields can inherit the picklist value set from a global picklist.

GlobalValueSet

Represents the metadata for a global picklist value set, which is the set of shared values that custom picklist fields can use. A global value set isn't a field itself. In contrast, the custom picklist fields that are based on a global picklist are of type ValueSet. This type extends the Metadata metadata type and inherits its `fullName` field.

GlobalValueSetTranslation

Contains details for a global value set translation. Global value sets are lists of values that can be shared by multiple custom picklist fields, optionally across objects. This type extends the Metadata metadata type and inherits its `fullName` field.

Group

Represents a set of public groups, which can have users, roles, and other groups.

HomePageComponent

Represents the metadata associated with a home page component. You can customize the Home tab in Salesforce Classic to include components such as sidebar links, a company logo, a dashboard snapshot, or custom components that you create. Use to create, update, or delete home page component definitions.

HomePageLayout

Represents the metadata associated with a home page layout. You can customize home page layouts and assign the layouts to users based on their user profile.

IdentityVerificationProcDef

Represents the definition of the identity verification process.

IdentityVerificationProcDtl

Represents the search functionality configuration and the minimum number of optional verifiers for identity verification. This type extends the Metadata metadata type and inherits its `fullName` field.

IdentityVerificationProcFld

Represents the search and verification fields used in identity verification. This type extends the Metadata metadata type and inherits its `fullName` field.

InboundCertificate

Represents a mutual authentication certificate that is imported to your Salesforce org.

InboundNetworkConnection

Represents a private connection between a third-party data service and a Salesforce org. The connection is inbound because the callouts are coming *into* Salesforce.This type extends the Metadata metadata type and inherits its `fullName` field.

IndustriesPricingSettings

Represents the settings for Salesforce Pricing.

IndustriesRatingSettings

Represents the settings for Rate Management.

InstalledPackage

Represents a first-generation managed package to be installed or uninstalled. Deploying a newer version of a currently installed package upgrades the package. To install an unlocked or second-generation managed package, use the `sf package install` Salesforce CLI command.

IntegrationProviderDef

Represents an integration definition associated with a service process. Stores data for the Industries: Send Apex Async Request and Industries: Send External Async Request invocable actions.

IPAddressRange

Represents a range of IP addresses to include in or exclude from the specified feature.

KeywordList

Represents a list of keywords used in Experience Cloud site moderation. This keyword list is a type of moderation criteria that defines offensive language or inappropriate content that you don't want in your site.

Layout

Represents the metadata associated with a page layout. For more information, see Page Layouts in Salesforce Help.

LearningItemType

Represents a custom exercise type that an Enablement user takes in an Enablement program in the Guidance Center. A custom exercise type also requires a corresponding LearningItem record for the Guidance Center and corresponding EnblProgramTaskDefinition and EnblProgramTaskSubCategory records for when admins create a program in Program Builder.

Letterhead

Represents formatting options for the letterhead in an email template. A letterhead defines the logo, page color, and text settings for your HTML email templates. Use letterheads to ensure a consistent look and feel in your company's emails.

LightningBolt

Represents the definition of a Lightning Bolt Solution, which can include custom apps, flow categories, and Experience Builder templates. This type extends the Metadata metadata type and inherits its `fullName` field.

LightningComponentBundle

Represents a Lightning web component bundle. A bundle contains Lightning web component resources.

LightningExperienceTheme

Represents the details of a custom theme, including the BrandingSet. Themes enable admins to specify configurable attributes, such as three colors and five images. The colors and some of the images override SLDS token values and influence the generation of `app.css`.

LightningMessageChannel

Represents the metadata associated with a Lightning Message Channel. A Lightning Message Channel represents a secure channel to communicate across UI technologies, such as Lightning Web Components, Aura Components, and Visualforce.

LightningOnboardingConfig

Represents the feedback provided when users switch from Lightning Experience to Salesforce Classic. Admins can customize the question, how frequently the form appears, and where the feedback is stored in Chatter from the Adoption Assistance page in Lightning Experience Setup. This type extends the Metadata metadata type and inherits its `fullName` field.

LightningTypeBundle

Represents a custom Lightning type. Use this type to override the default user interface to create a customized appearance of responses on the custom agent's action input and output. Deploy this bundle to your organization to implement the overrides.

LiveChatAgentConfig

Represents the configuration of an organization's Chat deployment, such as how many chats can be assigned to an agent and whether chat sounds are enabled.

LiveChatButton

Represents a Chat deployment's settings for the button that customers click to chat with an agent and the chat window, such as the label that appears on the button and the pre-chat form that appears before a chat begins. This type extends the Metadata metadata type and inherits its `fullName` field.

LiveChatDeployment

Represents the configuration settings for a specific Chat deployment, such as the branding image for the deployment and whether or not chat transcripts are automatically saved.

LiveChatSensitiveDataRule

Represents a rule for masking or deleting data of a specified pattern. Written as a regular expression (regex).

LoyaltyProgramSetup

Represents the configuration of a loyalty program process including its parameters and rules. Program processes determine how new transaction journals are processed. When new transaction journals meet the criteria and conditions for a program process, actions that are set up in the process are triggered for the transaction journals.

ManagedContentType

Represents the definition of custom content types for use with Salesforce CMS. Custom content types are displayed as forms with defined fields.

ManagedEventSubscription (Beta)

Represents a managed event subscription in Pub/Sub API. Use a managed event subscription to track the events that a subscriber client consumed and resume a subscription where it left off. This type extends the metadata type and inherits its `fullName` field.

ManagedTopics

Represents navigational and featured topics managed in an Experience Cloud site.

MarketingAppExtension

Represents an integration with a third-party app or service that is used to work with prospects.

MatchingRule

Represents a matching rule that is used to identify duplicate records.

MessagingChannel

Represents the metadata associated with an Embedded Service Messaging channel.

Metadata

The base class for all metadata types. You can't edit this object. A component is an instance of a metadata type.

MetadataWithContent

MetadataWithContent is the base type for all metadata types that contain content, such as documents or email templates. It extends Metadata. You can't edit this object.

MfgProgramTemplate

Represents a definition of a program to create a program-based business. A program-based business, also known as a Manufacturing Program, enables manufacturers to drive their business models with forecasting tools and manage the end-to-end sales process efficiently.

MilestoneType

Represents the name and description of a milestone, which you can use in an entitlement process to track important steps in cases.

MlDomain

Represents an Einstein Intent Set. This type extends the Metadata metadata type and inherits its `fullName` field.

MLDataDefinition

Represents a modeling data definition, which specifies the data used to create a model. Such data can include filters, fields to include, fields to exclude, and so on. This type extends the Metadata metadata type and inherits its `fullName` field.

MLPredictionDefinition

Represents a prediction definition that specifies details about the prediction. This type extends the Metadata metadata type and inherits its `fullName` field.

MobileApplicationDetail

Represents the packaging attributes for a mobile connected app. This type extends the Metadata metadata type and inherits its `fullName` field.

MobileSecurityAssignment

Represents the assignment of mobile app security policies to a profile. The policies apply to the Salesforce mobile app with Enhanced Mobile App Security enabled.

MobileSecurityPolicy

Represents a mobile app security policy on the Salesforce mobile app with Enhanced Mobile App Security enabled. For a full description of each policy, see Enable and Configure Mobile App Security Policies.

MobSecurityCertPinConfig

Represents the authentication server certificate pin configuration on the Salesforce mobile app with Enhanced Mobile Security.

ModerationRule

Represents a rule used in your Experience Cloud site to moderate member-generated content. Each rule specifies the member-generated content the rule applies to, the criteria to enforce the rule on, and the moderation action to take. Moderation rules help protect your site from spammers, bots, and offensive or inappropriate content. This type extends the Metadata metadata type and inherits its `fullName` field.

MutingPermissionSet

Represents a set of disabled permissions and is used in conjunction with PermissionSetGroup.

MyDomainDiscoverableLogin

Represents the configuration settings when the My Domain login page type is Discovery. Login Discovery provides an identity-first login experience, where the login page contains the identifier field only. Based on the identifier entered, a handler determines how to authenticate the user. This type extends the Metadata metadata type and inherits its `fullName` field.

NamedCredential

Represents a named credential, which specifies the URL of a callout endpoint and its required authentication parameters in one definition. A named credential can be specified as an endpoint to simplify the setup of authenticated callouts.

NavigationMenu

Represents the navigation menu in an Experience Builder site. A navigation menu consists of items that users can click to go to other parts of the site. This type replaces the NavigationLinkSet subtype on Network. NavigationMenu is available in API version 47.0 and later. This type extends the Metadata metadata type and inherits its `fullName` field.

Network

Represents an Experience Cloud site. Salesforce Experience Cloud lets you create branded spaces for your employees, customers, and partners. You can customize and create experiences, whether they're communities, sites, or portals, to meet your business needs, then transition seamlessly between them. If you want to create zones that contain Chatter Answers and Ideas, use the Community (Zone) component.

NetworkBranding

Represents the branding and color scheme applied to the login pages of an Experience Cloud site. (Experience Cloud sites are represented by the Network component.)

NotificationTypeConfig

Represents the metadata associated with org-level notification settings for standard and custom notification types. This type extends the Metadata metadata type and inherits its `fullName` field.

OauthCustomScope

Represents a permission defining the protected data that a connected app can access from an external entity when Salesforce is the OAuth authorization provider. This type extends the Metadata metadata type and inherits its `fullName` field.

OauthTokenExchangeHandler

Represents a token exchange handler. The token exchange handler also consists of an Apex class. During the OAuth 2.0 token exchange flow, the token exchange handler is used to validate tokens from an external identity provider and to map users to Salesforce.

OcrSampleDocument

Represents the details of a sample document or a document type that's used as a reference while extracting and mapping information from a customer form. This type extends the Metadata metadata type and inherits its `fullName` field.

OcrTemplate

Represents the details of the mapping between a form and a Salesforce object using Intelligent Form Reader.This type extends the Metadata metadata type and inherits its `fullName` field.

OmniExtTrackingDef

Represents a connection between an OmniTrackingGroup in OmniAnalytics and a third-party Analytics system such as Google Analytics.

OmniInteractionAccessConfig

Represents configuration settings for access to Omnistudio FlexCard caching and data sources.

OmniInteractionConfig

Represents configuration settings for Omnistudio.

OmniScript

Represents an OmniScript for the Discovery Framework, which guides users through sales, service, and other business processes. For Discovery Framework, the customization type is `discoveryframework`.

OmniSupervisorConfig

Represents the Omni-Channel supervisor configuration for an assigned group of supervisors.

OmniTrackingGroup

Represents a group of FlexCard and OmniScript components that have their user interactions tracked together in OmniAnalytics.

OutboundNetworkConnection

Represents a private connection between a Salesforce org and a third-party data service. The connection is outbound because the callouts are going *out* of Salesforce. This type extends the Metadata metadata type and inherits its `fullName` field.

Package

Specifies which metadata components to retrieve as part of a `retrieve()` call or defines a package of components.

ParticipantRole

Represents details, such as the name and associated default access level, for a role that a participant can have in the context of a parent record.

PathAssistant

Represents Path records.This type extends the Metadata metadata type and inherits its `fullName` field.

PaymentGatewayProvider

Represents the metadata associated with a payment gateway provider. This type extends the Metadata metadata type and inherits its `fullName` field.

PermissionSet

Represents a set of permissions that's used to grant more access to one or more users without changing their profile or reassigning profiles. You can use permission sets to grant access but not to deny access.

PermissionSetGroup

Represents a group of permission *sets* and the permissions within them. Use permission set groups to organize permissions based on job functions or tasks. Then, you can package the groups as needed.

PermissionSetLicenseDefinition (Developer Preview)

Represents the definition of a custom permission set license, which entitles specified features in a package.

PersonAccountOwnerPowerUser

Represents a user who can own more than 50,000 customer or partner portal accounts. Person account owner power users can own a large number of either customer or partner users. They can't change their role, look up to a parent role, or reparent their role. Person account owner power user objects can't be created if deferred sharing is turned on for your org. This object is available in API version 57.0 and later.

PipelineInspMetricConfig

Represents the settings of Pipeline Inspection forecast category metrics.

PlatformCachePartition

Represents a partition in the Platform Cache. This type extends the Metadata metadata type and inherits its `fullName` field.

PlatformEventChannel

Represents a channel that you can subscribe to in order to receive a stream of events. In API version 46.0 and earlier, it is the default standard channel for change data capture events. In API version 47.0 and later, it is a custom channel for change data capture events.

PlatformEventChannelMember

Represents an entity selected for Change Data Capture notifications on a standard or custom channel, or a platform event selected on a custom channel.

PlatformEventSubscriberConfig

Represents configuration settings for a platform event Apex trigger, including the batch size and the trigger's running user.

Portal

The Portal metadata type represents a partner portal.

PortalDelegablePermissionSet

Represents the org-level permission sets that can be assigned to a particular profile for external users or shoppers in a store after enabling the Delegable Administration perm.

PostTemplate

Represents the metadata associated with an approval post template for Approvals in Chatter. With approval post templates, you can customize the information included in approval request posts that appear in Chatter feeds. This type extends the Metadata metadata type and inherits its `fullName` field.

ProductAttributeSet

Represents the ProductAttribute information being used as and attribute such as color_*c, size*_c .

PresenceDeclineReason

Represents an Omni-Channel decline reason that agents can select when declining work requests. This type extends the Metadata metadata type and inherits its `fullName` field.

PresenceUserConfig

Represents a configuration that determines a presence user's settings.

PricingActionParameters

Represents the pricing action that's associated with a context definition and pricing procedure.

PricingRecipe

Represents the data models or sets of objects of a particular cloud that the pricing data store consumes during design time and run time.

Profile

Represents a user profile. A profile defines a user's permission to perform different functions within Salesforce. This type extends the Metadata metadata type and inherits its `fullName` field.

ProfileActionOverride

Represents an override of an ActionOverride by a user profile. You can use it to override an ActionOverride on a standard Home tab or object record page in Lightning Experience. When a user logs in with a profile, a matching ProfileActionOverride assignment takes precedence over existing overrides for the Home tab or record page specified in ActionOverride. In API versions 39.0 to 44.0, you can access ProfileActionOverride by accessing its encompassing CustomApplication on page 623 or Profile on page 1616 metadata types. In API version 45.0 and later, you can access ProfileActionOverride only by accessing its encompassing CustomApplication on page 623.

ProfilePasswordPolicy

Represents a profile's password policies. Profile password policies override org-wide password policies for that profile's users. Use ProfilePasswordPolicy to retrieve password policies for a given profile. This type extends the Metadata metadata type and inherits its `fullName` field.

ProfileSessionSetting

Represents a profile's session settings. Use ProfileSessionSetting to retrieve the session settings for a given profile. This type extends the Metadata metadata type and inherits its `fullName` field.

Prompt

Represents the metadata related to in-app guidance, which includes prompts and walkthroughs. Help users discover your products and services, adopt your processes, or learn how to use a new feature. Write the content, select the target audience, and specify where and when the in-app guidance appears.

PublicKeyCertificate

Represents the public key certificate. On this entity we store a public certificate or a JSON web key, which is used to validate the customer-provided JWT.

PublicKeyCertificateSet

Represents a set of public certificate keys. On this entity we store a public certificates or JSON web keys.

Queue

Represents a holding area for items before they are processed.

QueueRoutingConfig

Represents the settings that determine how work items are routed to agents.

QuickAction

Represents a specified create or update quick action for an object that then becomes available in the Chatter publisher. For example, you can create an action that, on the detail page of an account, allows a user to create a contact related to that account from the Chatter feed on that page. QuickAction can be created on objects that permit custom fields.

RedirectWhitelistUrl

Represents a trusted URL that's excluded from redirection restrictions when the `redirectionWarning` or `redirectBlockModeEnabled` field on the SessionSettings Metadata type is set to `true`. This type extends the Metadata metadata type and inherits its `fullName` field.

RecommendationStrategy

Represents a recommendation strategy. Recommendation strategies are applications, similar to data flows, that determine a set of recommendations to be delivered to the client through data retrieval, branching, and logic operations.

RecordActionDeployment

Represents configuration settings for the Actions & Recommendations, Action Launcher, and Bulk Action Panel components. For example, you can have a deployment that specifies which types of actions to display, default actions for channels, and the actions that users can add at runtime. If the component shows Next Best Action recommendations, the deployment configures which strategies to use and how recommendations appear. This type extends the Metadata metadata type and inherits its `fullName` field.

RecordAggregationDefinition

Represents a data aggregation from one object to another object to which it is connected by other objects in the data model.

RecordAlertCategory

Represents a category to group and present record alerts.

RegisteredExternalService

Represents a registered external service, which provides an extension or integration.

ReferencedDashboard

Represents the ReferencedDashboard object in CRM Analytics. A referenced dashboard stores information about an externally referenced dashboard.

RelatedRecordAssocCriteria

Represents criteria for automatically linking records like accounts, leads, opportunities, and cases with the branches that work with them.

RelationshipGraphDefinition

Represents a definition of a graph that you can configure in your organization to traverse object hierarchies and record details, giving you a glimpse of how your business works.

RemoteSiteSetting

Represents a remote site setting. Before any Visualforce page, Apex callout, or JavaScript code using XmlHttpRequest in an s-control or custom button can call an external site, that site must be registered in the Remote Site Settings page, or the call fails.

Report

Represents a custom report. This metadata type only supports custom reports; standard reports aren't supported.

ReportType

Represents the metadata associated with a custom report type. Custom report types allow you to build a framework from which users can create and customize reports.

RestrictionRule

Represents a restriction rule or a scoping rule. A restriction rule has `enforcementType` set to `Restrict` and controls the access that specified users have to designated records. A scoping rule has `enforcementType` set to `Scoping` and controls the default records that your users see without restricting access. This type extends the Metadata metadata type and inherits its `fullName` field.

Role

Represents a role in your organization.

RoleOrTerritory

Represents the common base type and valid values for role or territory.

SalesWorkQueueSettings

Represents settings used to customize work queue options for third-party scoring. In Sales Engagement, you can add a custom number field on person accounts, contacts, or leads. Then, use the custom number field to sort the work queue. This type extends the Metadata metadata type and inherits its `fullName` field.

SamlSsoConfig

Represents a SAML Single Sign-On configuration. This type extends the Metadata metadata type and inherits its `fullName` field. Single sign-on (SSO) is an authentication method that enables users to access multiple applications with one login and one set of credentials. For example, after users log in to your org, they can automatically access all apps from the App Launcher. You can set up your Salesforce org to trust a third-party identity provider to authenticate users. Or you can configure a third-party app to rely on your org for authentication.

SchedulingObjective

Represents a scheduling objective in Workforce Engagement. Scheduling objectives define business goals that the scheduling tools consider when identifying agents for shifts.

SchedulingRule

Represents a scheduling rule in Workforce Engagement Management. Scheduling rules determine when agents are assigned to shifts.

Scontrol

Deprecated. Represents an Scontrol component, corresponding to an s-control in the Salesforce user interface.

SearchCustomization

Represents the configuration of search settings created in Search Manager. The configuration includes the search channel, searchable objects and fields, and rules to filter search results.

SearchOrgWideObjectConfig

Represents an object in the search index. The search index contains org-wide search settings created in Search Manager. Each object in the search index includes searchable fields and fields protected by field-level security in search.

ServiceAISetupDefinition

Represents settings for an Einstein for Service feature such as Einstein Article Recommendations. This type extends the Metadata metadata type and inherits its `fullName` field.

ServiceAISetupField

Represents a field on cases or knowledge articles that Einstein uses to identify relevant articles in Einstein Article Recommendations. This type extends the Metadata metadata type and inherits its `fullName` field.

ServiceChannel

Represents a channel of work items that are received from your organization—for example, cases, chats, or leads.

ServicePresenceStatus

Represents a presence status that can be assigned to a service channel. This type extends the Metadata metadata type and inherits its `fullName` field.

ServiceProcess

Represents a process created in Service Process Studio and its associated attributes.

Settings

Represents the organization settings related to a feature. For example, your password policies, session settings and network access controls are all available in the SecuritySettings component type.

SharedTo

SharedTo defines the sharing access for a list view or a folder. It can be used to specify the target and source for owner-based sharing rules.

SharingBaseRule

Represents sharing rule settings such as access level and to whom access is granted.

SharingRules

Represents the base container for sharing rules, which can be criteria-based, ownership-based, territory-based, or for guest user access. SharingRules enables you to share records with a set of users, using rules that specify the access level for the target user group.

SharingSet

Represents a sharing set. A sharing set defines an access mapping that grants portal or community users access to objects that are associated with their accounts or contacts.

SiteDotCom

Represents a site for deployment.

Skill

Represents the settings for a skill used for field service or to route chats to agents in Chat, such as the name of the skill and which agents the skills are assigned to.

StandardValueSet

Represents the set of values in a standard picklist field. This type extends the Metadata metadata type and inherits its `fullName` field.

StandardValueSetTranslation

Contains details for a standard picklist translation. It returns a translated standard value set.This type extends the Metadata metadata type and inherits its `fullName` field.

StaticResource

Represents a static resource file, often a code library in a ZIP file. Static resources allow you to upload content that you can reference in a Visualforce page, including archives (such as .zip and .jar files), images, style sheets, JavaScript, and other files. Static resources can be used only within your Salesforce org, so you can't host content here for other apps or websites.

SustainabilityUom

Represents the unit of measure (UOM) values for custom fuel types in an org. Track fuel consumption and emission results with the flexibility to add custom fuel types and UOM values.

SustnUomConversion

Represents information about the unit of measure (UOM) conversion for the custom fuel types defined by a customer in an org.

SvcCatalogCategory

Represents the grouping of individual catalog items in Service Catalog.

SvcCatalogFulfillmentFlow

Represents the flow associated with a specific catalog item in the Service Catalog.

SvcCatalogItemDef

Represents the entity associated with a specific, individual service available in the Service Catalog.

SynonymDictionary

Represents a set of synonym groups, which are groups of words or phrases that are treated as equivalent in users' searches. You can define synonym groups to optimize search results for acronyms, variations of product names, and other terminology unique to your organization.

Territory

Represents a territory.

Territory2

Represents the metadata associated with a sales territory. This type extends the Metadata metadata type and inherits its `fullName` field. Available if Sales Territories has been enabled.

Territory2Model

Represents the metadata associated with a territory model in Sales Territories. This type extends the Metadata metadata type and inherits its `fullName` field.Available if Sales Territories has been enabled.

Territory2Rule

Represents the metadata associated with a territory assignment rule associated with an object, such as Account. Available if Sales Territories has been enabled.

Territory2Type

Represents the metadata for a category of territories in Sales Territories. Every Territory2 must have a Territory2Type. This type extends the Metadata metadata type and inherits its `fullName` field.Available if Sales Territories has been enabled.

TimelineObjectDefinition

Represents the container that stores the details of a timeline configuration. You can use this resource with Salesforce objects to see their records' related events in a linear time-sorted view.

TimeSheetTemplate

Represents a template for creating time sheets in Field Service. This type extends the Metadata metadata type and inherits its `fullName` field.

TopicsForObjects

Represents the ability to assign topics to objects or to remove topic assignments.

TransactionSecurityPolicy

Represents a transaction security policy definition. Transaction security policies give you a way to look through events in your organization and specify actions to take when certain combinations occur.

Translations

Metadata type that enables work with translations for various supported languages. The ability to translate component labels is part of the Translation Workbench.

UiFormatSpecificationSet

Represents a set of rules that define the style and visibility of conditional field formatting on Dynamic Forms-enabled Lightning page field instances.

UIObjectRelationConfig

Represents the admin-created configuration of the object relation UI component.

UserAccessPolicy

Represents a user access policy.

UserAuthCertificate

Represents a PEM-encoded user certificate. These certificates are associated with a user, and externally uploaded. The uploaded certificate is used to authenticate the user.

UserCriteria

Represents the member criteria to use in Experience Cloud site moderation rules. This type extends the Metadata metadata type and inherits its `fullName` field..

UserProfileSearchScope

Reserved for internal use.

UserProvisioningConfig

Represents information to use during a user provisioning request flow, such as the attributes for an update. This type extends the Metadata metadata type and inherits its `fullName` field.

VirtualVisitConfig

Represents an external video provider configuration, which relays events from Salesforce to the provider.

WaveAnalyticAssetCollection

Represents a collection of Analytics assets. This type extends the Metadata metadata type and inherits its `fullName` field.

WaveApplication

Represents the Analytics application. This type extends the Metadata metadata type and inherits its `fullName` field.

WaveComponent

Represents the WaveComponent object in the Analytics application. This type extends the MetadataWithContent metadata type and inherits its `content` and `fullName` fields.

WaveDataflow

Represents the WaveDataflow object in the Analytics application. This type extends the MetadataWithContent metadata type and inherits its `content` and `fullName` fields.

WaveDashboard

Represents the WaveDashboard object in the Analytics application. This type extends the MetadataWithContent metadata type and inherits its `content` and `fullName` fields.

WaveDataset

Represents the WaveDataset object in the Analytics application. This type extends the Metadata metadata type and inherits its `fullName` field.

WaveLens

Represents the WaveLens object in the Analytics application.

WaveRecipe

Represents the WaveRecipe type in an Analytics application. A recipe is a saved set of steps to perform on a specific source dataset or connected data. This type extends the MetadataWithContent metadata type and inherits its `content` and `fullName` fields.

WaveTemplateBundle

Represents an Analytics template bundle, which can be used to create Analytics apps. A bundle contains an Analytics template definition and all its related resources.This type extends the Metadata metadata type and inherits its `fullName` field.

WaveXmd

Represents the WaveXmd object in the Analytics application. This type extends the Metadata metadata type and inherits its `fullName` field.

WebStoreBundle

For internal use only.

WebStoreTemplate

Represents a configuration for creating commerce stores.

Workflow

Represents the metadata associated with a workflow rule. A workflow rule sets workflow actions into motion when its designated conditions are met. You can configure workflow actions to execute immediately when a record meets the conditions in your workflow rule, or set time triggers that execute the workflow actions on a specific day. Use this metadata type to create, update, or delete workflow rule definitions.

WorkSkillRouting

Represents a setup object that stores a set of WorkSkillRoutingAttribute objects. These objects are used to route a work item to an agent who has the skills necessary to take the work. This type extends the Metadata metadata type and inherits its `fullName` field.

Metadata Components and Types

Metadata components are not based on sObjects, like objects in the API. Instead, they are based on metadata types, such as ApexClass and CustomObject, which extend Metadata, the base class for all metadata types. A component is an instance of a metadata type.

For example, `CustomObject` is a metadata type for custom objects, and the `MyCustomObject__c` component is an instance of a custom object.

A metadata type can be identified in the metadata WSDL as any complexType that extends the Metadata complexType. A complexType that is a metadata type includes the following element in its WSDL definition:

```
<xsd:extension base="tns:Metadata">
```

CustomObject and BusinessProcess extend Metadata so they are metadata types; ActionOverride doesn't extend Metadata so it's not a metadata type.

You can individually deploy or retrieve a component for a metadata type. For example, you can retrieve an individual BusinessProcess component, but you can't retrieve an individual ActionOverride component. You can only retrieve an ActionOverride component by retrieving its encompassing CustomObject component.

Metadata components can be manipulated by asynchronous Metadata API calls or declarative (or file-based) Metadata API calls.

Most of the components can be accessed using Salesforce Extensions for Visual Studio Code. Exceptions are noted in the description of the object.

Field Data Types

Each component field has a specific field type. These field types can correspond to other components defined in the WSDL, or primitive data types, like `string`, that are commonly used in strongly typed programming languages.

These field data types are used in the messages that are exchanged between your client application and the API. When writing your client application, follow the data typing rules defined for your programming language and development environment. Your development tool handles the mapping of typed data in your programming language with these data types.

For more information, see Primitive Data Typesin the Salesforce Object Reference.

Enumeration Fields

Some component fields have a data type that is an enumeration. An enumeration is the API equivalent of a picklist. The valid values of the field are restricted to a strict set of possible values, all having the same data type. These values are listed in the field description column for each enumeration field. See `sortBy` for an example of an enumeration field of type string. The XML below shows a sample definition of an enumeration of type string in the WSDL.

```
<xsd:simpleType name="DashboardComponentFilter">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="RowLabelAscending"/>
        <xsd:enumeration value="RowLabelDescending"/>
        <xsd:enumeration value="RowValueAscending"/>
        <xsd:enumeration value="RowValueDescending"/>
    </xsd:restriction>
</xsd:simpleType>
```

Supported Calls

All of the metadata types are supported by the main calls, unless it is stated otherwise in the individual component sections. The main Metadata API calls are:

- CRUD calls, such as createMetadata() and deleteMetadata()
- File-based calls, such as deploy() and retrieve()
- Utility calls, such as listMetadata() and describeMetadata()

Metadata Coverage Report

Launch the Metadata Coverage report to determine supported metadata components. The Metadata Coverage report is the ultimate source of truth for metadata coverage across several channels. These channels include Metadata API, scratch org source tracking, unlocked packages, second-generation managed packages, classic managed packages, and more.

To view the Metadata Coverage report, you don't have to be logged into an org.

Unsupported Metadata Types

Some Salesforce features have metadata types that aren't available in Metadata API. These metadata types can't be retrieved or deployed with Metadata API. To make changes to these types, you must do it manually in each of your organizations.

Some metadata types may also be unsupported in source tracking, packaging, and change sets.

For a complete list of metadata types and where they're supported, see Metadata Coverage.

SEE ALSO:

Salesforce Developers: Metadata Coverage

Salesforce DX Developer Guide: Track Changes Between Your Project and Org

Second-Generation Managed Packaging Developer Guide: Second-Generation Managed Packages

Sandboxes: Staging Environments for Customizing and Testing: Change Sets

Special Behavior in Metadata API Deployments

Important considerations for specific types and contents of a deployment.

Use the information here to determine what to include in your deployment and how the changes appear in the destination.

Special Behavior in Deployments

Data Cloud Metadata Types

Check out the metadata types that are used for development in Data Cloud.

ActivationPlatform
Represents the ActivationPlatform configuration, such as platform name, delivery schedule, output format, and destination folder.

ActivationPlatformActvAttr
Represents the information about activation attributes. Reserved for future use.

ActivationPlatformField
Represents the information about the fields used in ActivationPlatform.

ActvPfrmDataConnectorS3
Represents the Amazon S3 bucket name and export directory.

ActvPlatformAdncIdentifier

Represents the information about the identifiers to be activated, such as Email, Phone, Mobile Advertiser (MAID) ID, and Over-the-top (OTT) ID.

ActvPlatformFieldValue

Represents the field values for the ActivationPlatformFields.

AiPluginUtteranceDef

Represents an utterance that can be used to pick a topic during runtime.

CustomerDataPlatformSettings

Represents an org's Data Cloud settings.

DataConnectorIngestApi

Represents the connection information specific to Ingestion API.

DataConnectorS3

Represents the connection information specific to Amazon S3.

DataPackageKitDefinition

Represents the top-level Data Kit container definition. Content objects can be added after the Data Kit is defined.

DataPackageKitObject

Represents the object in Data Kit Content Object. These objects are added inside the data kit.

DataSource

Used to represent the system where the data was sourced. This object is always needed when creating a Data Stream Definition.

DataSourceBundleDefinition

Represents the bundle of streams that a user adds to a data kit.

DataSourceField

Represents the details of a data source field.

DataSourceObject

Represents the object from where the data was sourced.

DataSourceTenant

For internal use only.

DataSrcDataModelFieldMap

Represents the entity that is used for storing the design time bundle level mappings for the data source fields and data model fields.

DataStreamDefinition

Contains data ingestion information such as connection, API, and file retrieval settings.

DataStreamTemplate

Represents the data stream that a user adds to a data kit.

ExternalDataConnector

Used to represent the object where the data was sourced.

ExternalDataSource

Represents the metadata associated with an external data source. Create external data sources to manage connection details for integration with data and content that are stored outside your Salesforce org.

ExternalDataTransportFieldTemplate

For internal use only.

ExternalDataTranObject

Represents a definition of a Data Cloud schema object. This type extends the Metadata metadata type and inherits its `fullName` field.

ExternalDataTransportObjectTemplate

For internal use only.

FieldSrcTrgtRelationship

Stores the relationships between a data model object (DMO) and its fields. For example, the `Individual.Id` field has a one-to-many relationship (1:M) with the `ContactPointEmail.PartyId` field.

InternalDataConnector

For internal use only.

MarketSegmentDefinition

Represents the field values for MarketSegmentDefinition. MarketSegmentDefinition is used to store the exportable metadata of a segment, such as segment criteria and other attributes. Developers can create segment definition packages, pass segment definition in the form of data build tool (DBT), and publish it on AppExchange for subscriber organizations to install and instantiate these segments.

MktCalcInsightObjectDef

Represents Calculated Insight definition such as expression.

MktDataTranObject

An entity that is used to deliver (aka transport) information from the source to a target (target will be called a landing entity).This can be the schema of a file, API, Event, or other means of transporting data, such as SubscriberFile1.csv, or SubscriberCDCEvent.

ObjectSourceTargetMap

Contains the object-level mappings between the source and the target objects. The source and target objects can be an MktDataLakeObject or an MktDataModelObject. For example, an Email source object can be mapped to the ContactPointEmail object.

StreamingAppDataConnector

Represents the connection information specific to Web and Mobile Connectors.

SEE ALSO:

Developer Center: Data Cloud

ActivationPlatform

Represents the ActivationPlatform configuration, such as platform name, delivery schedule, output format, and destination folder.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

ActivationPlatform components have the suffix `.activationPlatform` and are stored in the `activationPlatforms` folder.

Version

ActivationPlatform components are available in API version 54.0 and later.

Special Access Rules

There are no additional access requirements that are specific to this type.

Fields

Field Name	Description
`activationPlatformConnectorType`	**Field Type** ActivationPlatformConnectorType (enumeration of type string) **Description** Reserved for future use.
`activationPlatformAdditionalMetadata`	**Field Type** string **Description** Provides additional details about the activation platform, including subscriber override options, partner supported IDs, and file output properties such as maximum file size, compression settings, and the maximum number of records per file.
`dataConnector`	**Field Type** string **Description** Reference to the ActvPfrmDataConnectorS3 metadata type, which contains S3 bucket and export directory information into which Data Cloud writes data.
`description`	**Field Type** string **Description** Required. The description for ActivationPlatform.
`enabled`	**Field Type** boolean **Description** Required. Indicates if ActivationPlatform is enabled (`true`) or not (`false`). The default is false.
`includeSegmentNames`	**Field Type** boolean

Field Name	Description
	Description Indicates whether to include the segment name in metadata (`true`) or not (`false`).
`logoUrl`	**Field Type** string **Description** URL of the logo for the activation channel destination.
`masterLabel`	**Field Type** string **Description** Required. The name for the activation channel destination.
`notes`	**Field Type** string **Description** Notes for this ActivationPlartform.
`outputFormat`	**Field Type** ActivationPlatformFileOutputFormat (enumeration of type string) **Description** Required. The output format of the file. Valid values are: • `CSV` • `JSON` • `PARQUET`
`outputGrouping`	**Field Type** ActivationPlatformFileOutputGrouping (enumeration of type string) **Description** Required. The grouping of the output. Valid values are: • `PER_ACCOUNT` • `PER_SEGMENT`

Field Name	Description
`periodicRefreshFrequecy`	**Field Type** ActivationPlatformPeriodicFullRefresh (enumeration of type string) **Description** Reserved for internal use.
`platformType`	**Field Type** ActivationPlatformType (enumeration of type string) **Description** Required. The type of the Activation Platform. Valid values are: • `Advertising` • `Analytics` • `Marketing` • `Publishing` • `Technology`
`refreshFrequency`	**Field Type** ActivationPlatformRefreshFrequency (enumeration of type string) **Description** Required. Indicates how often the activation platform accepts data delivery. Valid value is: • `TWENTY_FOUR`
`refreshMode`	**Field Type** ActivationPlatformRefreshMode (enumeration of type string) **Description** Required. Defines how the refresh method handles refreshing files. Valid values are: • `FULL` • `INCREMENTAL`

Declarative Metadata Sample Definition

The following is an example of an ActivationPlatform component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ActivationPlatform xmlns="http://soap.sforce.com/2006/04/metadata">
    <dataConnector>S3Connector</dataConnector>
    <description>Activation Platform Description</description>
    <enabled>false</enabled>
    <includeSegmentNames>false</includeSegmentNames>
    <logoUrl>link to logo</logoUrl>
    <masterLabel>MyExternalPlatform</masterLabel>
    <notes>Notes about this Platform</notes>
    <outputFormat>CSV</outputFormat>
    <outputGrouping>PER_ACCOUNT</outputGrouping>
    <refreshMode>FULL</refreshMode>
    <refreshFrequency>TWENTY_FOUR</refreshFrequency>
    <periodicRefreshFrequecy>NEVER</periodicRefreshFrequecy>
    <platformType>Advertising</platformType>
</ActivationPlatform>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <fullName>MyActivationPlatform</fullName>
    <types>
        <members>APlatform</members>
        <name>ActivationPlatform</name>
    </types>
    <types>
        <members>AccountIdField</members>
        <name>ActivationPlatformField</name>
    </types>
    <types>
        <members>S3Connector</members>
        <name>ActvPfrmDataConnectorS3</name>
    </types>
    <types>
        <members>EmailIdentifier</members>
        <name>ActvPlatformAdncIdentifier</name>
    </types>
    <types>
        <members>AccountIdFieldValue</members>
        <name>ActvPlatformFieldValue</name>
    </types>
    <version>54.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ActivationPlatformActvAttr

Represents the information about activation attributes. Reserved for future use.

ActivationPlatformField

Represents the information about the fields used in ActivationPlatform.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

ActivationPlatformField components have the suffix `.activationPlatformField` and are stored in the `activationPlatformFields` folder.

Version

ActivationPlatformField components are available in API version 54.0 and later.

Special Access Rules

There are no additional access requirements that are specific to this type.

Fields

Field Name	Description
`activationPlatform`	**Field Type** string **Description** Required. Reference to the ActivationPlatform metadata type.
`helpText`	**Field Type** string **Description** Information about ActivationPlatformField.
`isHidden`	**Field Type** boolean

Field Name	Description
	Description Required. Indicates whether ActivationPlatformField can be overridden by marketer (`false`) or not (`true`). The default is false. Field can't be overridden by marketer when set to `true`.
`isRequired`	**Field Type** boolean **Description** Required. Indicates whether this ActivationPlatformField is required (`true`) or not (`false`). The default is false.
`masterLabel`	**Field Type** string **Description** Required. The name of the ActivationPlaformField.
`type`	**Field Type** ActivationPlatformFieldDataType (enumeration of type string) **Description** Represents the datatype of the field. Valid value is: • `Text`

Declarative Metadata Sample Definition

The following is an example of an ActivationPlatformField component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ActivationPlatformField xmlns="http://soap.sforce.com/2006/04/metadata">
    <activationPlatform>APlatform</activationPlatform>
    <isHidden>false</isHidden>
    <isRequired>true</isRequired>
    <masterLabel>AccountId</masterLabel>
</ActivationPlatformField>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <fullName>MyActivationPlatform</fullName>
```

```
    <types>
        <members>APlatform</members>
        <name>ActivationPlatform</name>
    </types>
    <types>
        <members>AccountIdField</members>
        <name>ActivationPlatformField</name>
    </types>
    <version>54.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ActvPfrmDataConnectorS3

Represents the Amazon S3 bucket name and export directory.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

ActvPfrmDataConnectorS3 components have the suffix `.actvPfrmDataConnectorS3` and are stored in the `actvPfrmDataConnectorS3s` folder.

Version

ActvPfrmDataConnectorS3 components are available in API version 54.0 and later.

Special Access Rules

There are no additional access requirements that are specific to this type.

Fields

Field Name	Description
`bucketName`	**Field Type** string **Description** Required.

Field Name	Description
	The Amazon S3 bucket name.
`exportDirectory`	**Field Type** string **Description** This is an optional field that is reserved for internal use.
`masterLabel`	**Field Type** string **Description** Required. The display name of ActvPfrmDataConnectorS3.

Declarative Metadata Sample Definition

The following is an example of an ActvPfrmDataConnectorS3 component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ActvPfrmDataConnectorS3 xmlns="http://soap.sforce.com/2006/04/metadata">
    <bucketName>MyS3Bucket</bucketName>
    <exportDirectory>Output</exportDirectory>
    <masterLabel>S3Connector</masterLabel>
</ActvPfrmDataConnectorS3>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <fullName>MyActivationPlatform</fullName>
    <types>
        <members>APlatform</members>
        <name>ActivationPlatform</name>
    </types>
    <types>
        <members>S3Connector</members>
        <name>ActvPfrmDataConnectorS3</name>
    </types>
    <version>54.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ActvPlatformAdncIdentifier

Represents the information about the identifiers to be activated, such as Email, Phone, Mobile Advertiser (MAID) ID, and Over-the-top (OTT) ID.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

ActvPlatformAdncIdentifier components have the suffix `.actvPlatformAdncIdentifier` and are stored in the `actvPlatformAdncIdentifiers` folder.

Version

ActvPlatformAdncIdentifier components are available in API version 54.0 and later.

Special Access Rules

There are no additional access requirements that are specific to this type.

Fields

Field Name	Description
`activationPlatform`	**Field Type** string **Description** Required. Reference to the ActivationPlatform metadata type.Reference to ActivationPlatform.
`identifierHashMethod`	**Field Type** ActivationPlatformIdentifierHashMethod (enumeration of type string) **Description** The hash method of the identifier type. The supported hash method for Email and Phone is `SHA256`. The supported hash method for MAID and OTT is `NONE`.
`identifierType`	**Field Type** ActivationPlatformIdentifierType (enumeration of type string) **Description** Required.

Field Name	Description
	The type of identifier to be activated. Valid values are: • EMAIL • MAID • OTT • PHONE
masterLabel	**Field Type** string **Description** Required. The name of the identifier.

Declarative Metadata Sample Definition

The following is an example of an ActvPlatformAdncIdentifier component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ActvPlatformAdncIdentifier xmlns="http://soap.sforce.com/2006/04/metadata">
    <activationPlatform>APlatform</activationPlatform>
    <identifierHashMethod>SHA256</identifierHashMethod>
    <identifierType>EMAIL</identifierType>
    <masterLabel>EmailIdentifier</masterLabel>
</ActvPlatformAdncIdentifier>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <fullName>MyActivationPlatform</fullName>
    <types>
        <members>APlatform</members>
        <name>ActivationPlatform</name>
    </types>
    <types>
        <members>EmailIdentifier</members>
        <name>ActvPlatformAdncIdentifier</name>
    </types>
    <version>54.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ActvPlatformFieldValue

Represents the field values for the ActivationPlatformFields.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

ActvPlatformFieldValue components have the suffix `.actvPlatformFieldValue` and are stored in the `actvPlatformFieldValues` folder.

Version

ActvPlatformFieldValue components are available in API version 54.0 and later.

Special Access Rules

Fields

Field Name	Description
`activationPlatformField`	**Field Type** string **Description** Required. Reference to the ActivationPlatform metadata type.
`isDefault`	**Field Type** boolean **Description** Required. Indicates whether the value is default (`true`) or not (`false`). The default is false. Picklist isn't supported in API version 54.0
`masterLabel`	**Field Type** string **Description** Required. The name of the field.

Field Name	Description
value	**Field Type** string **Description** The value of activationPlatformField.

Declarative Metadata Sample Definition

The following is an example of an ActvPlatformFieldValue component.

Field with no value:

```
<ActvPlatformFieldValue xmlns="http://soap.sforce.com/2006/04/metadata">
    <activationPlatformField>AccountIdField</activationPlatformField>
    <isDefault>true</isDefault>
    <masterLabel>AccountIdValue</masterLabel>
    <value>null</value>
</ActvPlatformFieldValue>
```

Field with value:

```
<ActvPlatformFieldValue xmlns="http://soap.sforce.com/2006/04/metadata">
    <activationPlatformField>AccountIdField</activationPlatformField>
    <isDefault>true</isDefault>
    <masterLabel>AccountIdValue</masterLabel>
    <value>1234</value>
</ActvPlatformFieldValue>
```

The following is an example package.xml that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <fullName>MyActivationPlatform</fullName>
    <types>
        <members>APlatform</members>
        <name>ActivationPlatform</name>
    </types>
    <types>
        <members>AccountIdField</members>
        <name>ActivationPlatformField</name>
    </types>
    <types>
        <members>AccountIdValue</members>
        <name>ActvPlatformFieldValue</name>
    </types>
    <version>54.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character * (asterisk) in the package.xml manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

AiPluginUtteranceDef

Represents an utterance that can be used to pick a topic during runtime.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

AiPluginUtteranceDef components are part of other components, such as GenAiPlugin, and aren't used separately.

Version

AiPluginUtteranceDef components are available in API version 63.0 and later.

Special Access Rules

AiPluginUtteranceDef is available only if Agents is enabled in your org.

Fields

Field Name	Description
`developerName`	**Field Type** string **Description** Required. Represents the API name of the utterance. Can contain only underscores and alphanumeric characters and must be unique in your org. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
`language`	**Field Type** string **Description** Required. The language of the utterance.
`masterLabel`	**Field Type** string **Description** Required. The master label for the utterance.

Field Name	Description
`utterance`	**Field Type** string **Description** Required. The utterance.

Declarative Metadata Sample Definition

See GenAiPlugin on page 1271.

CustomerDataPlatformSettings

Represents an org's Data Cloud settings.

Parent Type and Manifest Access

This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all the settings metadata types for the org are accessed using the "Settings" name. See Settings for more details.

File Suffix and Directory Location

`CustomerDataPlatformSettings` values are stored in the `CustomerDataPlatformSettings.settings` file in the `settings` folder. The `.settings` files are different from other named components, because there is only one settings file for each settings component.

Version

CustomerDataPlatformSettings components are available in API version 48.0 and later.

Special Access Rules

Fields

Field Name	Description
`enableCustomerDataPlatform`	**Field Type** boolean **Description** Indicates whether an org has Data Cloud enabled (`true`) or not (`false`).

Declarative Metadata Sample Definition

The following is an example of a CustomerDataPlatformSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<CustomerDataPlatformSettings xmlns="http://soap.sforce.com/2006/04/metadata">
  <enableCustomerDataPlatform>true</enableCustomerDataPlatform>
</CustomerDataPlatformSettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
  <types>
    <members>CustomerDataPlatform</members>
    <name>Settings</name>
  </types>
  <version>55.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

DataConnectorIngestApi

Represents the connection information specific to Ingestion API.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

DataConnectorIngestApi components have the suffix `.dataConnectorIngestApi` and are stored in the `dataConnectorIngestApis` folder.

Version

DataConnectorIngestApi components are available in API version 54.0 and later.

Special Access Rules

You must have the CustomizeApplication user permissions to access the DataConnectorIngestApi type.

Fields

Field Name	Description
masterLabel	**Field Type** string **Description** Required. UI label of the Ingestion API Connector.
sourceName	**Field Type** string **Description** Required. Name of the Ingestion API Connector.

Declarative Metadata Sample Definition

The following is an example of a DataConnectorIngestApi component.

```
<?xml version="1.0" encoding="UTF-8"?>
<DataConnectorIngestApi xmlns="http://soap.sforce.com/2006/04/metadata">
    <sourceName>CONNECTOR NAME</sourceName>
    <masterLabel>CONNECTOR NAME</masterLabel>
</DataConnectorIngestApi>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<DataConnectorIngestApi xmlns="http://soap.sforce.com/2006/04/metadata">
    <sourceName>MyConnector</sourceName>
    <masterLabel>MyConnector</masterLabel>
</DataConnectorIngestApi>
```

Wildcard Support in the Manifest File

This metadata type doesn't support the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

DataConnectorS3

Represents the connection information specific to Amazon S3.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

DataConnectorS3 components have the suffix s3DataConnector and are stored in the s3DataConnectors folder.

Version

DataConnectorS3 components are available in API version 50.0 and later.

Special Access Rules

You need the Salesforce CustomizeApplication permission to access this object.

Fields

Field Name	Field Type	Description
`fileNameWildcard`	string	Optional. File or Wildcard (*) to be used when finding files.
`importFromDirectory`	string	Required. Path from the directory to where files are located.
`masterLabel`	string	Required. The UI name for the S3 data connector.
`s3BucketName`	string	Optional. The Amazon S3 Name of the Bucket.

Declarative Metadata Sample Definition

The following is an example of a DataConnectorS3 component.

```
<?xml version="1.0" encoding="UTF-8"?>
<DataConnectorS3 xmlns="http://soap.sforce.com/2006/04/metadata">
<fileNameWildcard>*.csv</fileNameWildcard>
<importFromDirectory>c360-subset-lheader/</importFromDirectory>
<masterLabel>Person</masterLabel>
<s3BucketName>bucketeer-aa32faea-8431-4635-8a1d-b323a2d66c7c</s3BucketName>
</DataConnectorS3>
```

DataPackageKitDefinition

Represents the top-level Data Kit container definition. Content objects can be added after the Data Kit is defined.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. Because changing terms in our code can break current implementations, we maintained this metadata type's name.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

`DataPackageKitDefinition` components have the suffix `.dataPackageKitDefinition` and are stored in the `dataPackageKitDefinitions` folder.

Version

DataPackageKitDefinition components are available in API version 53.0 and later.

Special Access Rules

There are no additional access requirements that are specific to this type.

Fields

Field Name	Description
`dataKitType`	**Field Type** string **Description** Represents the type of data kit created. Available in API version 63.0 and later. Valid values are: • Default • Sandbox
`dataKitSource`	**Field Type** string **Description** Represents the data source in the updated data kit. Available in API version 63.0 and later.
`description`	**Field Type** string **Description** The description of the data kit.
`developerName`	**Field Type** string **Description** Required. Represents the name of the application. Can contain only underscores and alphanumeric characters and must be unique in your org. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.

Field Name	Description
isDeployed	**Field Type** boolean **Description** Indicates whether Data Kit content is deployed.
isEnabled	**Field Type** boolean **Description** Indicates whether the Data Kit is enabled.
masterLabel	**Field Type** string **Description** Required. Label that identifies the AI application throughout the Salesforce user interface.
versionNumber	**Field Type** double **Description** Auto incremented version number.
dataSpaceDefinitionDevName	**Field Type** string **Description** Data space name used to create the data kit. Available in API version 61.0 and later.
deploymentOrder	**Field Type** string **Description** Deployment order of components that are added to the data kit. Available in API version 61.0 and later.

Declarative Metadata Sample Definition

The following is an example of a DataPackageKitDefinition component.

```
<DataPackageKitDefinition xmlns="http://soap.sforce.com/2006/04/metadata">
    <developerName>SalesforceCRM</developerName>
    <isDeployed>false</isDeployed>
    <isEnabled>false</isEnabled>
    <masterLabel>SalesforceCRM</masterLabel>
    <versionNumber>1.0</versionNumber>
</DataPackageKitDefinition>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <fullName>SalesforceDataKit</fullName>
    <types>
        <members>SalesforceCRM</members>
        <name>DataPackageKitDefinition</name>
    </types>
    <types>
        <members>Admin</members>
        <name>Profile</name>
    </types>
    <version>53.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

DataPackageKitObject

Represents the object in Data Kit Content Object. These objects are added inside the data kit.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. Because changing terms in our code can break current implementations, we maintained this metadata type's name.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

`DataPackageKitObject` components have the suffix `.DataPackageKitObject` and are stored in the `DataPackageKitObjects` folder.

Version

DataPackageKitDefinition components are available in API version 53.0 and later.

Special Access Rules

There are no additional access requirements that are specific to this type.

Fields

Field Name	Description
masterLabel	**Field Type** string **Description** Required. Label that identifies the AI application throughout the Salesforce user interface.
parentDataPackageKitDefinitionName	**Field Type** string **Description** Required. Name of the data kit definition
referenceObjectName	**Field Type** string **Description** Required. The name of the data kit content.
referenceObjectType	**Field Type** string **Description** Required. The type of the content object in the data kit.

Declarative Metadata Sample Definition

The following is an example of a DataPackageKitDefinition component.

```
<?xml version="1.0" encoding="UTF-8"?>
<DataPackageKitObject xmlns="http://soap.sforce.com/2006/04/metadata">
    <masterLabel>CRM</masterLabel>
    <parentDataPackageKitDefinitionName>CRM</parentDataPackageKitDefinitionName>
    <referenceObjectName>CRM</referenceObjectName>
    <referenceObjectType>DLO</referenceObjectType>
</DataPackageKitObject>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <fullName>SalesforceDataKit</fullName>
    <types>
        <members>CRM</members>
        <name>DataPackageKitObject</name>
    </types>
    <types>
        <members>Admin</members>
```

```
        <name>Profile</name>
    </types>
    <version>53.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

DataSource

Used to represent the system where the data was sourced. This object is always needed when creating a Data Stream Definition.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

DataSource components have the suffix `dataSource` and are stored in the `mktDataSources` folder.

Version

DataSource components are available in API version 50.0 and later.

Special Access Rules

You need the Salesforce CustomizeApplication permission to access this object.

Fields

Field Name	Field Type	Description
`masterLabel`	string	Required. The UI name for the Data Source.
`prefix`	string	Required. Prefix for the Data Source to make Data Source Object records unique.

DataSourceBundleDefinition

Represents the bundle of streams that a user adds to a data kit.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

`DataSourceBundleDefinition` components have the suffix `.dataSourceBundleDefinition` and are stored in the `dataSourceBundleDefinitions` folder.

Version

DataSourceBundleDefinition components are available in API version 52.0 and later.

Special Access Rules

You need Data Cloud permission to access this object.

Fields

Field Name	Description
`dataPlatform`	**Field Type** string **Description** Required. Indicates the connector type that the streams in the bundle belong to.
`description`	**Field Type** string **Description** A description of the associated data source bundle. This field is available in API version 53.0 and later.
`icon`	**Field Type** string **Description** The icon used in the deployment flow. This field is available in API version 53.0 and later.
`isMultiDeploymentSupported`	**Field Type** boolean **Description** Indicates if the bundle can be deployed multiple times or not. Default value is `false`.
`masterLabel`	**Field Type** string **Description** Required. Indicates the name of the bundle.

Field Name	Description
bundleVersion	**Field Type** int **Description** Indicates the version of the bundle. This field is available in API version 63.0 and later.

Declarative Metadata Sample Definition

The following is an example of a DataSourceBundleDefinition component.

```
<?xml version="1.0" encoding="UTF-8"?>
<DataSourceBundleDefinition xmlns="http://soap.sforce.com/2006/04/metadata">
    <dataPlatform>Salesforce_Sales_and_Service_Cloud</dataPlatform>
    <isMultiDeploymentSupported>true</isMultiDeploymentSupported>
    <bundleVersion>1</bundleVersion>
    <masterLabel>b2</masterLabel>
</DataSourceBundleDefinition>
```

The following is an example `package.xml` that references the previous definition.

```
<types>
      <members>b2</members>
      <name>DataSourceBundleDefinition</name>
</types>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

DataSourceField

Represents the details of a data source field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

`DataSourceField` components have the suffix `.dataSourceField` and are stored in the `dataSourceFields` folder.

Version

DataSourceField components are available in API version 52.0 and later.

Special Access Rules

You need the Salesforce Customize Application permission to access this metadata type.

Fields

Field Name	Field Type	Description
`datatype`	string	Required. Indicates the data type of the field: text, number, or date.
`dateFormat`	string	The date format of date, time, date/time fields.
`definitionCreationType`	DefinitionCreationType (enumeration of type string)	Describes whether this object was added by the user or as part of a standard taxonomy. Available in API version 62.0 and later. Valid values are: • `Activation_Audience` (Reserved for internal use only) • `ADG` • Activation_Audience. Available in API version 63.0 and later. • `Bridge` • `Calculated_Insight` • `CG_Audience` • `Chunk` • `Curated` • `Custom` • `Derived` • `Directory_Table` • `External` • `Ml_Prediction` • `Segment_Membership` • `Semantic` • `Standard` • `System` • `Transform` • `Vector_Embedding`
`externalDataType`	string	The type of data in the external system. Available in API version 63.0 and later.
`externalName`	string	Required. Name of the object in the external system. This is different from the developer name.
`fieldFormula`	string	Used for formulas.
`isDataRequired`	boolean	If `true`, data is required. Default value is `false`.
`isEventDate`	boolean	If `true`, an event date is required. Default value is `false`. Available in API version 63.0 and later.
`isFormula`	boolean	If `true`, a formula is required. Default value is `false`.

Field Name	Field Type	Description
isRecordModified	boolean	If true, the system tracks the modification date of the record. Default value is false. Available in API version 63.0 and later.
keyQualifierName	string	Contains the developer name of the keyQualifier field. Available in API version 62.0 and later.
length	int	Length of a string column.
masterLabel	string	Required. Field label.
precision	int	The total number of digits in a number including decimal points. Used for currency and for numeric accuracy.
primaryIndexOrder	int	If supplied, indicates that this field is part of the primary key. The number value indicates the order of attributes if it's a compound primary key. A missing value means that this field isn't part of the primary key.
scale	int	The number of digits to the right of the decimal point. Used for currency and for numeric accuracy.
sequence	int	Required. The sequence of this source schema.
srcKeyQualifier	string	Contains a reference to the source key qualifier record. The source key is the MktDataLakeSrcKeyQualifer metadata type. Available in API version 55.0 and later.
usageTag	usageTag (enumeration of type string)	Indicates if the field is a key qualifier field. Available in API version 55.0 and later. Values are: • KEY_QUALIFIER—The field is used as a key qualifier field. • NONE—The field isn't used as a key qualifier field.
versionNumber	double	Required. The version of the data source object.

Declarative Metadata Sample Definition

This is an example of a DataSourceObject component and its fields. You can retrieve the DataSourceField component only through its parent object, DataSourceObject.

```
<DataSourceObject xmlns="http://soap.sforce.com/2006/04/metadata"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <dataSource>test1108</dataSource>
    <dataSourceFields>
        <fullName>DOB</fullName>
        <datatype>D</datatype>
        <externalName>DOB</externalName>
        <isDataRequired>false</isDataRequired>
        <masterLabel>DOB</masterLabel>
        <sequence xsi:nil="true"/>
        <versionNumber xsi:nil="true"/>
    </dataSourceFields>
```

```
    <externalRecordIdentifier>individuals_20200125_000000_csv</externalRecordIdentifier>
    <masterLabel>test1108</masterLabel>
    <objectType>Object</objectType>
</DataSourceObject>
```

The following is an example `package.xml` that references the previous definition.

```
<types>
        <members>test1108</members>
        <name>DataSource</name>
    </types>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

DataSourceObject

Represents the object from where the data was sourced.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

DataSourceObject components have the suffix `dataSourceObject` and are stored in the `mktDataSourceObjects` folder.

Version

DataSourceObject components are available in API version 50.0 and later.

Special Access Rules

You need the Salesforce Customize Application permission to access this metadata type.

Fields

Field Name	Field Type	Description
`accelerationEnabled`	AccelerationEnabled (enumeration of type string)	Acceleration of data stream processing. Available in API version 63.0 and later. Possible values are: • `Yes` • `No`

Field Name	Field Type	Description
additionalDLOInfoJson	string	Additional information about the Directory Table data lake object (DLO), such as the directory path and file pattern.
creationType	string	Indicates whether this object was added by the user or as part of a standard taxonomy.
dataConnection	string	The source data connector for the Directory Table DLO.
dataSource	string	Required. The system where the data was sourced.
dataSourceFields	DataSourceField[] on page 186	An array of data source fields.
dmoDeveloperName	string	The developer name of the Directory Table data model object (DMO).
dmoLabel	string	The UI name of the Directory Table DMO.
externalDatabaseName	string	The name of the external database for the data source object. Available in API version 63.0 and later.
externalObjectName	string	The external data source object. Available in API version 63.0 and later.
externalRecordIdentifier	string	The identifier for the data source.
externalSchemaName	string	The name of the schema within the external database. Available in API version 63.0 and later.
masterLabel	string	Required. The UI name for the data source object.
objectCategory	string	The category of the data source object.
objectType	DataObjectType (enumeration of type string)	The object type. Possible values are: • API • Object • Table
sourceObject	string	The developer name of the source object for the data source object.
storageType	StorageType (enumeration of type string)	The type of storage used for data source object. Available in API version 63.0 and later. Possible values are: • External • Local
templateVersion	int	The version number of the data source object.

DataSourceTenant

For internal use only.

DataSrcDataModelFieldMap

Represents the entity that is used for storing the design time bundle level mappings for the data source fields and data model fields.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

`DataSrcDataModelFieldMap` components have the suffix `.dataSrcDataModelFieldMap` and are stored in the `dataSrcDataModelFieldMaps` folder.

Version

DataSrcDataModelFieldMap components are available in API version 53.0 and later.

Special Access Rules

You need Data Cloud permission to access this object.

Fields

Field Name	Description
`masterLabel`	**Field Type** string **Description** Required. Indicates the name of the entity.
`sourceField`	**Field Type** string **Description** Required. Indicates the developer name of data source fields.
`targetField`	**Field Type** string **Description** Required. Indicates the developer name of data mapping object fields.
`versionNumber`	**Field Type** double

Field Name	Description
	Description Required. Indicates the version number of the entity.

Declarative Metadata Sample Definition

The following is an example of a DataSrcDataModelFieldMap component.

```
<?xml version="1.0" encoding="UTF-8"?>
<DataSrcDataModelFieldMap xmlns="http://soap.sforce.com/2006/04/metadata">
    <masterLabel>DataSrcDataModel26</masterLabel>
    <sourceField>Account1.LastModifiedDate__c</sourceField>
    <targetField>ssot__Account__dlm.ssot__LastModifiedDate__c</targetField>
    <versionNumber>1.0</versionNumber>
</DataSrcDataModelFieldMap>
```

The following is an example `package.xml` that references the previous definition.

```
<types>
        <members>DataSrcDataModel26</members>
        <name>DataSrcDataModelFieldMap</name>
</types>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

DataStreamDefinition

Contains data ingestion information such as connection, API, and file retrieval settings.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

DataStreamDefinition components have the suffix `dataStreamDefinition` and are stored in the `dataStreamDefinitions` folder.

Version

DataStreamDefinition components are available in API version 50.0 and later.

Special Access Rules

You need the Salesforce CustomizeApplication permission to access this object.

Fields

Field Name	Field Type	Description
areHeadersIncludedInFile	boolean	If true, headers are included in the file if this is a single file stream.
bulkIngest	boolean	If true, files are aggregated before data is ingested if the file names contain a wildcard. For example, `profiles*.csv`.
definitionCreationType	string	Required. Enum tracks the source of an object or field creation. Valid values are: • Custom • Standard Valid values available in API version 62.0 and later are: • ADG • Calculated_Insight • CG_Audience • Chunk • Directory_Table • External • Semantic • Transform • Vector_Embedding
dataConnector	string	Required. Describe whether this data stream definition was created by a customer or by an internal system.
dataConnectorType	DataConnectorType (enumeration of type string)	The ingestion data source. Valid values are: • ACCOUNTENGAGEMENT • AwsS3 • AzureBlob • BIG_QUERY • CuratedEntity • DataCloud • ExternalPlatform • GoogleCloudStorage • IngestApi • REDSHIFT • SalesforceCommerceCloud • SalesforceDotCom • SalesforceInteractionStudio • SalesforceMarketingCloud • SFTP

Field Name	Field Type	Description
		• `Snowflake` • `StreamingApp` • `UPLOAD`
`dataExtensionIdentifier`	string	For a Marketing Cloud data extension, the unique identifier.
`dataExtractField`	string	Name of the transport field that's used when the extract method is CDC.
`dataExtractMethods`	DataImportDataExtractMethods (enumeration of type string)	Describes how to identify the data to be extracted. Valid values include: • `DATETIME_CDC` • `FULL_REFRESH` • `NUMERIC_CDC` • `BINARY_CDC` (reserved for future use)
`dataPlatDataSetBundle`	string	Identifies which data set bundle this definition was created from. Useful in cases where the same item can be configured across data connections.
`dataPlatformDataSet`	string	The description is provided by the developer.
`dataPlatformDataSetItemName`	string	Name of the Data Platform Set Item.
`dataSource`	string	Required. A reference to the data source from which the data originated. This is usually the API name or a unique system identifier, such as the enterprise ID (EID) of the customer in the format `MC_<EID>`. Example: `MCEnterprise` or `MC_12345`.
`description`	string	Required. A description of the data stream definition.
`fileNameWildcard`	string	File or wildcard (*) used when finding files.
`internalOrganization`	string	The name of the internal organization.
`isLimitedToNewFiles`	boolean	If true, file retrieval is limited to new files.
`isMissingFileFailure`	boolean	If true, treat the case of missing files as a failure.
`masterLabel`	string	Required. UI label for this data stream definition.
`mktDataLakeObject`	string	Required. Reference to the landing entity (target) where data will be stored.
`mktDataTranObject`	string	Reference to the object that's used to transport information from the source to a landing entity (target).

DataStreamTemplate

Represents the data stream that a user adds to a data kit.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

`DataStreamTemplate` components have the suffix `.dataStreamTemplate` and are stored in the `dataStreamTemplates` folder.

Version

DataStreamTemplate components are available in API version 53.0 and later.

Special Access Rules

You need the Salesforce Customize Application permission to access this metadata type.

Fields

Field Name	Description
`dataConnectionSourceParameters`	**Field Type** DataConnectionParamTmpl[] **Description** Input representation for the data connection source parameters.
`dataSourceBundleDefinition`	**Field Type** string **Description** Required. Reference to the bundle to which this template belongs.
`dataSourceObject`	**Field Type** string **Description** Required. Reference to the Data Source Objects (DSOs). A DSO represents the object from where the data was sourced.
`filterCriteria`	**Field Type** string **Description** Filter applied to the data stream before the information is sent to Data Cloud.
`masterLabel`	**Field Type** string **Description** Required. Name assigned to the data stream template.

Field Name	Description
`objectCategory`	**Field Type** string **Description** Required. Category of the Data Model Object (DMO).
`refreshDayOfMonth`	**Field Type** int **Description** The duration of the day of the month after which the data stream must be refreshed.
`refreshDayOfWeek`	**Field Type** int **Description** The duration of the day of the week after which the data stream must be refreshed.
`refreshFrequency`	**Field Type** DataImportRefreshFrequency (enumeration of type string) **Description** The frequency with which the datastream must be refreshed. Possible values are: • `Batch` • `NONE` • `MINUTES_15` • `MINUTES_30` • `MINUTES_5` • `HOURLY` • `DAILY` • `WEEKLY` • `MONTHLY` • `NOT_APPLICABLE` • `STREAMING`
`refreshHours`	**Field Type** string **Description** The duration after which the datastream must be refreshed.
`refreshMode`	**Field Type** DataImportRefreshMode (enumeration of type string) **Description** The mode of refresh. Possible values are:

Field Name	Description
	• FULL_REFRESH • UPSERT • INCREMENTAL • REPLACE • NEAR_REAL_TIME_INCREMENTAL • NOT_APPLICABLE • PARTIAL_UPDATE • STREAMING
refreshStartDate	**Field Type** date **Description** The date to retrieve data based on the refresh frequency data. Available in API version 62.0 and later.
sourceObjectName	**Field Type** string **Description** The name of the source object from which data is streamed. Available in API version 62.0 and later.
streamType	**Field Type** StreamType (enumeration of type string) **Description** The type of data stream. Available in API version 62.0 and later. Possible values are: • DIRECT_ACCESS • DIRECT_ACCESS_ACCELERATED • INGEST
streamingAppDataConnectorType	**Field Type** streamingAppDataConnectorType (enumeration of type string) **Description** The connector app for data streaming. Available in API version 63.0 and later. Possible values are: • MobileApp • WebApp
templateVersion	**Field Type** int **Description** The version number of the template. Available in API version 62.0 and later.

Declarative Metadata Sample Definition

The following is an example of a DataStreamTemplate component.

```
<DataStreamTemplate xmlns="http://soap.sforce.com/2006/04/metadata">
    <dataSourceBundleDefinition>b2</dataSourceBundleDefinition>
    <dataSourceObject>sd3ds</dataSourceObject>
    <masterLabel>b2</masterLabel>
    <objectCategory>Profile</objectCategory>
</DataStreamTemplate>
```

The following is an example `package.xml` that references the previous definition.

```
 <types>
        <members>ssd3s</members>
        <name>DataStreamTemplate</name>
    </types>
```

Wildcard Support in the Manifest File

This metadata type doesn't support the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ExternalDataConnector

Used to represent the object where the data was sourced.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

ExternalDataConnector components have the suffix externalDataConnector and are stored in the externalDataConnectors folder.

Version

ExternalDataConnector components are available in API version 50.0 and later.

Special Access Rules

You need the Salesforce CustomizeApplication permission to access this object.

Fields

Field Name	Field Type	Description
`dataConnectionStatus`	DataConnectionStatus (enumeration of type string)	Indicates whether you're connected to a data source. Valid values are: • `Connected` • `Disconnected` • `Failed`

Field Name	Field Type	Description
dataConnectorConfiguration	string	Reference to the Data Connector Configuration that is used to retrieve or receive data such as DataConnectorS3.
dataConnectorType	DataConnectorType (enumeration of type string)	Type of connection such as AmazonS3. Valid values are: • ACCOUNTENGAGEMENT • AmazonS3 • CuratedEntity • DataCloud • ExternalPlatform • GoogleCloudStorage • IngestApi • SalesforceCommerceCloud • SalesforceDotCom • SalesforceInteractionStudio • SalesforceMarketingCloud • SFTP • StreamingApp • UPLOAD
dataPlatform	string	Reference to the Data Platform that provides or uses this data, such as Amazon_S3.
externalDataTranObjects	ExternalDataTranObject	Stores the schema objects related to the data connector. Available in API version 56.0 and later.
masterLabel	string	Required. The UI name for the ExternalDataConnector.

Declarative Metadata Sample Definition

The following is an example of a ExternalDataConnector component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ExternalDataConnector xmlns="http://soap.sforce.com/2006/04/metadata">
<dataConnectionStatus>Connected</dataConnectionStatus>
<dataConnectorConfiguration>Person</dataConnectorConfiguration>
<dataConnectorType>AmazonS3</dataConnectorType>
<dataPlatform>Amazon_S3</dataPlatform>
<masterLabel>AmazonS3</masterLabel>
</ExternalDataConnector>
```

ExternalDataSource

Represents the metadata associated with an external data source. Create external data sources to manage connection details for integration with data and content that are stored outside your Salesforce org.

Note: All credentials stored within this entity are encrypted under a framework that is consistent with other encryption frameworks on the platform. Salesforce encrypts your credentials by auto-creating org-specific keys. Credentials encrypted using the previous encryption scheme have been migrated to the new framework.

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

ExternalDataSource components are stored in the `dataSources` directory of the corresponding package directory. ExternalDataSource components have the suffix `.dataSource`, and the prefix is the name of the external data source.

Version

ExternalDataSource components are available in API version 28.0 and later.

Special Access Rules

As of Spring '20 and later, only authenticated internal and external users can access this type.

Fields

Field Name	Field Type	Description
`authProvider`	string	The authentication provider represented by the AuthProvider component.
`certificate`	string	If you specify a certificate, your Salesforce org supplies it when establishing each two-way SSL connection with the external system. The certificate is used for digital signatures, which verify that requests are coming from your Salesforce org. Tip: For best performance, verify that your remote HTTPS encrypted sites have OCSP (Online Certificate Status Protocol) stapling turned on.
`customConfiguration`	string	A string of configuration parameters that are specific to the external data source's `type`. • customConfiguration for Salesforce Connect—Cross-Org Adapter • customConfiguration for Salesforce Connect—OData 2.0 or 4.0 Adapter • customConfiguration for Salesforce Connect—Custom Adapter
`customHttpHeaders`	CustomHttpHeaders[]	Represents custom HTTP headers used with OData 2.0 or OData 4.0 connectors. Available in API version 43.0 or later.

Field Name	Field Type	Description
`endpoint`	string	The URL of the external system, or if that URL is defined in a named credential, the named credential URL. Corresponds to `URL` in the user interface. A named credential URL contains the scheme `callout:`, the name of the named credential, and an optional path. For example: `callout:My_Named_Credential/some_path`. You can append a query string to a named credential URL. Use a question mark (?) as the separator between the named credential URL and the query string. For example: `callout:My_Named_Credential/some_path?format=json`.
`externalDataSrcDescriptors`	ExternalDataSrcDescriptors[]	Represents schema descriptors for an external data source used with Salesforce Connect adapter for Amazon DynamoDB (available in API version 55.0 or later) or Amazon Athena (available in API version 56.0 or later).
`isWritable`	boolean	Lets the Lightning Platform and users in this org create, update, and delete records for external objects associated with the external data source. The external object data is stored outside the org. By default, external objects are read only. Corresponds to `Writable External Objects` in the user interface. Available in API version 35.0 and later. However, with the cross-org adapter for Salesforce Connect, you can set this field to `true` only in API version 39.0 and later.
`label`	string	A user-friendly name for the external data source. The label is displayed in the Salesforce user interface, such as in list views. Examples include Acme Team Marketing Site, or Acme SharePoint.
`namedCredential`	string	Represents the definition of the referenced named credential for an external data source of the type Amazon DynamoDB or Amazon Athena.
`oauthRefreshToken`	string	The OAuth refresh token. Used to obtain a new access token for an end user when a token expires.
`oauthScope`	string	Specifies the scope of permissions to request for the access token. Corresponds to `Scope` in the user interface.
`oauthToken`	string	The access token issued by the external system.
`password`	string	The password to be used by your org to access the external system. Ensure that the credentials you use have adequate privileges to access the external system, perform searches, return data, and return information about the external system's metadata.

Field Name	Field Type	Description
`principalType`	`External PrincipalType` (enumeration of type string)	Determines whether you're using one set or multiple sets of credentials to access the external system. Corresponds to `Identity Type` in the user interface. The valid values are: • `Anonymous` • `PerUser` • `NamedUser`
`protocol`	`Authentication Protocol` (enumeration of type string)	The authentication protocol that's required to access the external system. The valid values are: • `NoAuthentication` • `Oauth` • `Password` For cloud-based Files Connect external systems, select **Oauth 2.0**. For on-premises systems, select **Password Authentication**. For Simple URL data sources, select **No Authentication**.
`repository`	string	Used for SharePoint Online. If metadata is not accessible, use this field to create tables and default table fields.
`type`	`ExternalData SourceType` (enumeration of type string)	Required. For Salesforce Connect, specifies the adapter that connects to the external system. The valid values are: • `AmazonAthena`—Amazon Athena • `AmazonDynamoDB`—Amazon DynamoDB • `OData`—OData 2.0 adapter • `OData4`—OData 4.0 adapter • `SfdcOrg`—cross-org adapter • *`ApexClassId`*—`DataSource.Provider` class that defines the custom adapter created via the Apex Connector Framework For Files Connect, specifies the data source type. The valid values are: • `ContentHubSharepoint`—SharePoint 2010 or 2013 • `ContentHubSharepointOffice365`—SharePoint Online • `ContentHubSharepointOneDrive`—OneDrive for Business • `ContentHubGDrive`—Google Drive • `ContenHubIsotope`—Isotope If Chatter is enabled, you can also specify `SimpleURL` to access data hosted on a web server that doesn't require authentication.

Field Name	Field Type	Description
		• `outgoingemail`—A data source used for sending an email through a quick action. The `Identity` and `Wrapper` types are reserved for future use. For Digital Lending Configurator, the valid value is `AFPPAttribute`. The data source name for the Application Form Product Proposal Attribute virtual object. For the federated search external data source type, the valid value is `OpenSearch`. For Transaction Management in Revenue Cloud, the valid values are: • `ASPAttribute`— The data source name for the Asset State Period Attribute virtual object. Available in API version 63.0 and later. • `OIAttribute`—The data source name for the Order Item Attribute virtual object. Available in API version 63.0 and later. • `QLIAttribute`—The data source name for the Quote Line Item Attribute virtual object. Available in API version 63.0 and later.
`username`	string	The username to be used by your org to access the external system. Ensure that the credentials you use have adequate privileges to access the external system, perform searches, return data, and return information about the external system's metadata.
`version`	string	Reserved for future use.

CustomHttpHeaders

Represents a custom HTTP header used with OData 2.0 or OData 4.0 connectors. Available in API version 43.0 or later.

Field Name	Field Type	Description
`description`	string	A text description of the header field's purpose.
`headerFieldName`	string	Required. Name of the header field. The name must contain at least one alphanumeric character or underscore. It can also include: ! # $ % & ' * + - . ^ _ ` | ~
`headerFieldValue`	string	Required. A formula that resolves to the value for the header. The values in the formula must evaluate to a string. If the formula resolves to null and an empty string, the header isn't sent.
`isActive`	boolean	Indicates whether the custom HTTP header is available to use (`true`) or unavailable (`false`).

`customConfiguration` for Salesforce Connect—Cross-Org Adapter

The following sample JSON-encoded configuration string defines parameters that apply when the external data source's `type` is set to `SfdcOrg`.

```
{"apiVersion":"32.0","environment":"CUSTOM",
"searchEnabled":"true","timeout":"120"}
```

The parameters correspond to these fields in the user interface:

- `apiVersion`—`API Version`
- `environment`—`Connect to`
- `searchEnabled`—`Enable Search`
- `timeout`—`Connection Timeout`

`customConfiguration` for Salesforce Connect—OData 2.0 or 4.0 Adapter

The following JSON-encoded configuration string defines parameters that apply when the external data source's `type` is set to `OData` or `OData4`.

```
{"inlineCountEnabled":"true","csrfTokenName":"X-CSRF-Token",
"requestCompression":"false","pagination":"CLIENT",
"noIdMapping":"false","format":"ATOM",
"searchFunc":"","compatibility":"DEFAULT",
"csrfTokenEnabled":"true","timeout":"120",
"searchEnabled":"true"}
```

The parameters correspond to these fields in the user interface.

- `compatibility`—`Special Compatibility`
- `csrfTokenEnabled`—`CSRF Protection`
- `csrfTokenName`—`Anti-CSRF Token Name`
- `format`—`Format`
- `inlineCountEnabled`—`Request Row Counts`
- `noIdMapping`—`High Data Volume`
- `pagination`—`Server Driven Pagination`
- `requestCompression`—`Compress Requests`
- `searchEnabled`—`Enable Search`
- `searchFunc`—`Custom Query Option for Salesforce Search`
- `timeout`—`Connection Timeout`

Declarative Metadata Sample Definition: OData 2.0 or 4.0

The following is the definition of an external data source for Salesforce Connect—OData 2.0 or 4.0 adapter.

```
<?xml version="1.0" encoding="UTF-8"?>
<ExternalDataSource xmlns="http://soap.sforce.com/2006/04/metadata">
    <authProvider>FacebookAuth</authProvider>
    <customConfiguration>{"compatibility":"DEFAULT",
    "noIdMapping":"false","inlineCountEnabled":"true",
    "searchEnabled":"true","format":"ATOM",
    "requestCompression":"false","pagination":"SERVER",
```

```
    "timeout":"120"}</customConfiguration>
    <customHttpHeaders>
        <headerFieldName>X-User</headerFieldName>
        <headerFieldValue>$User.Username</headerFieldValue>
    </customHttpHeaders>
    <endpoint>http://myappname.herokuapp.com/DataHub.svc</endpoint>
    <label>DataHub</label>
    <principalType>NamedUser</principalType>
    <protocol>Oauth</protocol>
    <type>OData</type>
</ExternalDataSource>
```

`customConfiguration` for Salesforce Connect—Custom Adapter

The following sample JSON-encoded configuration string defines the parameter that applies when the external data source's `type` is set to the ID of a `DataSource.Provider` class.

```
{"noIdMapping":"false"}
```

The `noIdMapping` parameter corresponds to the `High Data Volume` field in the user interface.

`ExternalDataSrcDescriptors` for Salesforce Connect Adapter for Amazon DynamoDB and for Amazon Athena

Represents schema descriptors for an external data source used with Salesforce Connect adapter for Amazon DynamoDB (available in API version 55.0 or later) or Amazon Athena (available in API version 56.0 or later).

Field Name	Field Type	Description
`customObject`	string	If set, the external object associated with the descriptor.
`descriptor`	string	Required. The descriptor document that contains the metadata information.
`descriptorVersion`	string	If the external system supports schema versioning for the data source, the optional descriptor document version tracks the external system's schema version. Several descriptors with different document versions may be active.
`developerName`	string	Required. The unique name of the child-level setup entity.
`externalDataSource`	string	Required. The name of the external data source associated with the descriptor.
`subtype`	ExternalDataSrcDescSubtype (enumeration of type string)	Required. The subtype of the descriptor. Values are: • `SchemaTableMetadata`— Used to cache information about the external system. • `SchemaTableQualifiers`— Used to customize the data retrieval query to the external system.

Field Name	Field Type	Description
systemVersion	int	Required. The version that defines the descriptor format and allows compatibility with descriptor formats between Salesforce releases.
type	ExternalDataSrcDescType (enumeration of type string)	Required. The type of the descriptor. Valid value: • Schema

Declarative Metadata Sample Definition: Amazon DynamoDB

The following is an example of an external data source for the Salesforce Connect adapter for Amazon DynamoDB that uses ExternalDataSrcDescriptor component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ExternalDataSource xmlns="http://soap.sforce.com/2006/04/metadata">
    <customConfiguration>{"timeout":"120"}</customConfiguration>
    <externalDataSrcDescriptors>
        <fullName>MyQualifierName</fullName>
        <customObject>MyExternalObject__x</customObject>
        <descriptor>
        {
          "tableName": "MyDynamoDBTable",
          "columns": {
            "MyField": {"presence": "exists"}
          }
        }
        </descriptor>
        <developerName>MyQualifierName</developerName>
        <externalDataSource>MyDataSource</externalDataSource>
        <subtype>SchemaTableQualifiers</subtype>
        <systemVersion>0</systemVersion>
        <type>Schema</type>
    </externalDataSrcDescriptors>
    <isWritable>true</isWritable>
    <label>MyDataSource</label>
    <namedCredential>MyNamedCredential</namedCredential>
    <principalType>Anonymous</principalType>
    <protocol>NoAuthentication</protocol>
    <type>AmazonDynamoDb</type>
</ExternalDataSource>
```

Declarative Metadata Sample Definition: Amazon Athena

The following is an example of an external data source for the Salesforce Connect adapter for Amazon Athena that uses ExternalDataSrcDescriptor component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ExternalDataSource xmlns="http://soap.sforce.com/2006/04/metadata">
    <customConfiguration>
    {
      "DataCatalog": "AwsDataCatalog",
      "timeout": "120"
```

```
    }
    </customConfiguration>
    <externalDataSrcDescriptors>
        <fullName>MyAthenaQualifierName</fullName>
        <customObject>MyAthenaExternalObject__x</customObject>
        <descriptor>
        {
          "tableName": "myathenadatabase.myathenatable",
          "extendedQualifiers": {"workgroup": "primary"},
          "keyColumns": ["ExternalIdComponent", "OtherExternalIdComponent"]
        }
        </descriptor>
        <developerName>MyAthenaQualifierName</developerName>
        <externalDataSource>MyAthenaDataSource</externalDataSource>
        <subtype>SchemaTableQualifiers</subtype>
        <systemVersion>0</systemVersion>
        <type>Schema</type>
    </externalDataSrcDescriptors>
    <isWritable>false</isWritable>
    <label>MyAthenaDataSource</label>
    <namedCredential>MyAthenaNamedCredential</namedCredential>
    <principalType>Anonymous</principalType>
    <protocol>NoAuthentication</protocol>
    <type>AmazonAthena</type>
</ExternalDataSource>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ExternalDataTransportFieldTemplate

For internal use only.

ExternalDataTranObject

Represents a definition of a Data Cloud schema object. This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

ExternalDataTranObject components have the suffix `.externalDataTranObject` and are stored in the `externalDataTranObjects` folder.

Version

ExternalDataTranObject components are available in API version 55.0 and later.

Special Access Rules

ExternalDataTranObject is available only if Data Cloud is provisioned.

Fields

Field Name	Description
`availabilityStatus`	**Field Type** AvailabilityStatus (enumeration of type string) **Description** Required. Represents the availability of the object. Valid values are: • `Available` • `In_Use`
`creationType`	**Field Type** DefinitionCreationType (enumeration of type string) **Description** Required. Describes whether this object was added by the Customer or as part of a Standard Taxonomy or by the System. Valid values are: • `Segment_Membership` • `Activation_Audience` (Reserved for internal use only) • `Custom` • `Standard` • `System` • `Derived` • `Bridge` • `Curated` • `Standard` Valid values available in API version 62.0 and later are: • `ADG` • `Calculated_Insight` • `CG_Audience` • `Chunk` • `Directory_Table` • `External` • `Semantic` • `Transform` • `Vector_Embedding`

Field Name	Description
`extDataTranObjectTemplate`	**Field Type** string **Description** Reserved for internal use and read-only. Reference to the associated ExtDataTranObjectTemplate data kit object. The system populates this field when a data kit that contains a data stream is deployed.
`externalDataTranFields`	**Field Type** ExternalDataTranField **Description** Optional. Stores the fields related to that schema object.
`masterLabel`	**Field Type** string **Description** Required. The UI name for the object.
`mktDataTranObject`	**Field Type** MktDataTranObject **Description** Optional. An entity that is used to transport information from the source to a target.
`objectCategory`	**Field Type** string **Description** Required. Reference to the Object Category. For Transport, they're Profile, Engagement, or Other.

ExternalDataTranField

Stores the fields related to ExternalDataTranObject schema.

Field Name	Description
`creationType`	**Field Type** DefinitionCreationType (enumeration of type string).

Field Name	Description
	Description Required. Describes whether this object was added by the Customer or as part of a Standard Taxonomy or by the System. Valid values are: • `Segment_Membership` • `Custom` • `Standard` • `System` • `Derived` • `Bridge` • `Curated`
`datatype`	**Field Type** string **Description** Required. Phone, currency, number, or other assigned type.
`dateFormat`	**Field Type** string **Description** Optional. The Date format of date, time, date/time fields in this Transport field.
`extDataTranFieldTemplate`	**Field Type** string **Description** Reserved for internal use and read-only. Reference to the associated ExtDataTranFieldTemplate data kit object. The system populates this field when a data kit that contains a data stream is deployed.
`externalName`	**Field Type** string **Description** Optional. Name of the object in the external system (different from Developer Name).
`isCurrencyIsoCode`	**Field Type** boolean

Field Name	Description
	Description Optional. If true, this field is a currency ISO code.
`isDataRequired`	**Field Type** boolean **Description** Optional. If true, data is required for this field.
`length`	**Field Type** int **Description** Optional. Length of a string column.
`masterLabel`	Optional. Field label.
`mktDataTranField`	**Field Type** mktDataTranFieldType on page 212 **Description** Optional.
`precision`	**Field Type** int **Description** Optional. Used for currency and numeric accuracy.
`primaryIndexOrder`	**Field Type** int **Description** Optional. If supplied, indicates this field is part of the primary key where the number value (starting at 1) indicates the order of attributes if it's a compound primary key. Missing value means this field isn't part of the primary key.
`scale`	**Field Type** int

Field Name	Description
	Description Optional. Used for currency and numbers.
`sequence`	**Field Type** int **Description** Optional. The sequence of this source schema.

MktDataTranField

Stores fields related to MktDataTranObject.

Field Name	Description
`creationType`	**Field Type** DefinitionCreationType (enumeration of type string). **Description** Required. Describes whether this object was added by the Customer or as part of a Standard Taxonomy or by the System. Valid values are: • `Segment_Membership` • `Custom` • `Standard` • `System` • `Derived` • `Bridge` • `Curated` • Valid values available in API version 62.0 and later are: – `ADG` – `Calculated_Insight` – `CG_Audience` – `Chunk` – `Directory_Table` – `External` – `Semantic` – `Transform`

Field Name	Description
	- `Vector_Embedding`
`datatype`	**Field Type** string **Description** Required. Phone, currency, number, or other assigned type.
`dateFormat`	**Field Type** string **Description** Optional. The Date format of date, time, date/time fields in this Transport field.
`externalName`	**Field Type** string **Description** Optional. Name of the object in the external system (different from Developer Name).
`isDataRequired`	**Field Type** boolean **Description** Optional. If true, data is required for this field.
`length`	**Field Type** int **Description** Optional. Length of a string column.
`masterLabel`	Optional. Field label.
`precision`	**Field Type** int **Description** Optional. Used for currency and numeric accuracy.

Field Name	Description
primaryIndexOrder	**Field Type** int **Description** Optional. If supplied, indicates this field is part of the primary key where the number value (starting at 1) indicates the order of attributes if it's a compound primary key. Missing value means this field isn't part of the primary key.
scale	**Field Type** int **Description** Optional. Used for currency and numbers.
sequence	**Field Type** int **Description** Optional. The sequence of this source schema.

Declarative Metadata Sample Definition

The following is an example of a ExternalDataTranObject component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ExternalDataTranObject xmlns="http://soap.sforce.com/2006/04/metadata">
    <CreationType>Custom</CreationType>
    <DataConnectorId>1WMxx00000000mPGAQ</DataConnectorId>
    <DeveloperName>Third_Identity_Create</DeveloperName>
    <Language>en_US</Language>
    <MasterLabel>Identity Create</MasterLabel>
    <ObjectCategoryId>0hqxx000000009hAAA</ObjectCategoryId>
    <Status>Available</Status>
</ExternalDataTranObject>
```

The following is an example `package.xml` that references the previous definition.

```
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>ExternalDataTranObject</name>
    </types>
```

```
    <version>55.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ExternalDataTransportObjectTemplate

For internal use only.

FieldSrcTrgtRelationship

Stores the relationships between a data model object (DMO) and its fields. For example, the `Individual.Id` field has a one-to-many relationship (1:M) with the `ContactPointEmail.PartyId` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

FieldSrcTrgtRelationship components have the suffix `.fieldSrcTrgtRelationship` and are stored in the `fieldSrcTrgtRelationships` folder.

Version

FieldSrcTrgtRelationship components are available in API version 51.0 and later.

Special Access Rules

To access this metadata type, you must have the Customize Application user permission.

Fields

Field Name	Field Type	Description
`definitionCreationType`	DefinitionCreationType (enumeration of type string)	Required. Describes whether this object was added by the user or as part of a standard taxonomy. Values are: • `ADG` • `Activation_Audience`

Field Name	Field Type	Description
		• `Bridge` • `Calculated_Insight` • `Chunk` • `Curated` • `Custom` • `Derived` • `Directory_Table` • `External` • `Ml_Prediction` • `Segment_Membership` • `Semantic` • `Standard` • `System` • `Transform` • `Vector_Embedding`
`lookupFieldName`	string	Reference to the DMO lookup field.
`masterLabel`	string	Required. The UI name for the field relationship.
`owner`	FieldSrcTrgtRelationshipOwner (enumeration of type string)	Optional. The type of relationship that exists between the source and the target. Values are: • `SObject` - The source of the relationship is a DMO and the target is a standard or custom SObject. • `DataCloud` - Both the source and the target of the relationship are DMOs. The field is needed only when the target is an SObject. The system can infer the value when the target is a DMO.
`relationshipCardinality`	RelationshipCardinality (enumeration of type string)	Required. Cardinality of the relationship between the source and target fields. Values are: • `ManyToOne` • `OneToOne`
`sourceFieldName`	string	Required. Name of the field that represents the source side of the relationship.
`targetEntity`	string	Optional. Name of the entity that represents the target side of the relationship. The target entity can be a DMO or a standard or custom SObject. The field is needed only when the target entity is an SObject. The system can infer the value when the target entity is a DMO.

Field Name	Field Type	Description
`targetFieldName`	string	Required. Name of the field that represents the target side of the relationship.

InternalDataConnector

For internal use only.

MarketSegmentDefinition

Represents the field values for MarketSegmentDefinition. MarketSegmentDefinition is used to store the exportable metadata of a segment, such as segment criteria and other attributes. Developers can create segment definition packages, pass segment definition in the form of data build tool (DBT), and publish it on AppExchange for subscriber organizations to install and instantiate these segments.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata type and inherits its `fullName` field.

File Suffix and Directory Location

`MarketSegmentDefinition` components have the suffix `.marketSegmentDefinition` and are stored in the `marketSegmentDefinitions` folder.

Version

MarketSegmentDefinition components are available in API version 55.0 and later.

Fields

Field Name	Description
`additionalMetadata`	**Field Type** string **Description** An XML clob to hold name value pairs for storing additional metadata. Not applicable for DBT type segment.
`excludeCriteria`	**Field Type** string **Description** Holds the JSON exclude criteria for UI based segments. Not applicable for DBT or Lookalike segments.

Field Name	Description
includeCriteria	**Field Type** string **Description** An XML wrapped in a CDATA section that captures DBT definition. Only single model DBT is supported.
masterLabel	**Field Type** string **Description** Required. Display name of the field value.
segmentOn	**Field Type** string **Description** Required when `segmentType` is `UI`. Points to relevant MktDataModelObject entity instance. Must be a valid developerName for an MktDataModelObject instance of Profile type.
segmentType	**Field Type** MarketSegmentType (enumeration of type string) **Description** Required. Type of the segment to be created. Only DBT is supported via API. Values are: • `DBT` • `Lookalike` • `UI` • `EinsteinGPTSegmentsUI`

Declarative Metadata Sample Definition

The following is an example of a MarketSegmentDefinition component.

```
<?xml version="1.0" encoding="UTF-8"?>
<MarketSegmentDefinition>
    <segmentType>DBT</segmentType>
    <includeCriteria>
    <![CDATA[
        <DbtPipeline>
            <models>
                <model>
                    <name>m1</name>
                    <sql>select ssot__Individual__dlm.ssot__Id__c from
```

```
ssot__Individual__dlm</sql>
                </model>
            </models>
        </DbtPipeline>
        ]]>
    </includeCriteria>
    <masterLabel>msd2_simple</masterLabel>
</MarketSegmentDefinition>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:Package xmlns:ns2="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>MarketSegmentDefinition</name>
    </types>
    <version>55.0</version>
</ns2:Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

MktCalcInsightObjectDef

Represents Calculated Insight definition such as expression.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

MktCalcInsightObjectDef components have the suffix mktCalcInsightObjectDef and are stored in the mktCalcInsightObjectDefs folder.

Version

MktCalcInsightObjectDef components are available in API version 52.0 and later.

Special Access Rules

You need the Salesforce CustomizeApplication permission to access this object.

Fields

Field Name	Field Type	Description
builderExpression	string	Reserved for internal use.
creationType	CalculatedInsightCreationType (enumeration of type string)	Required. Describes whether this Calculated Insight Object Definition was added was added by the customer. Valid values include: Custom.
description	string	The description for this Calculated Insight Object Definition.
expression	string	Required when the Calculated Insight Object Definition is for internal insight type. This is the SQL query to generate the calculated insight.
masterLabel	string	Required. App name for this Calculated Insight Object Definition.
system	string	Required. Indicates how this calculated insight object definition was added, by the customer or by the system. Valid values are: • Custom • System (API version 61.0 and later)

Declarative Metadata Sample Definition

The following is an example of a MktCalcInsightObjectDef component.

```
<?xml version="1.0" encoding="UTF-8"?>
<MktCalcInsightObjectDef xmlns="http://soap.sforce.com/2006/04/metadata">
    <creationType>Custom</creationType>
    <description>InsightName description</description>
    <expression>SELECT COUNT(ssot__Individual__dlm.ssot__Id__c) as count__c FROM
ssot__Individual__dlm</expression>
</MktCalcInsightObjectDef>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character * (asterisk) in the package.xml manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

MktDataTranObject

An entity that is used to deliver (aka transport) information from the source to a target (target will be called a landing entity).This can be the schema of a file, API, Event, or other means of transporting data, such as SubscriberFile1.csv, or SubscriberCDCEvent.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

MktDataTranObject components have the suffix mktDataTranObject and are stored in the mktDataTranObjects folder.

Version

MktDataTranObject components are available in API version 50.0 and later.

Special Access Rules

You need the Salesforce CustomizeApplication permission to access this object.

Fields

Field Name	Field Type	Description
`connector`	string	Required. Data Cloud connector name that allows you to connect the data source to Data Cloud.
`creationType`	DefinitionCreationType	Required. Describe whether this object was added as the result of the Customer or as part of a Standard Taxonomy. • Valid values available in API version 62.0 and later are: – `ADG` – `Calculated_Insight` – `CG_Audience` – `Chunk` – `Directory_Table` – `External` – `Semantic` – `Transform` – `Vector_Embedding`
`dataSource`	string	Required. Your reference to the data source from which the data originated (source of that data such as the name of a CRM Org. Example: MC Enterprise.
`dataSourceObject`	string	Required. Represents the object name from where the data is sourced. Example: ecommerce-OrderItem.
`masterLabel`	string	Required. The UI name for the Data Transport Object.
`objectCategory`	string	Required. Reference to the Object Category. For Transport, these are Profile, Engagement, or Other.

MktDataTranField

This is a sub-type to MktDataTranObject.

Field Name	Field Type	Description
`creationType`	DefinitionCreationType	Optional: Was this object added as a result of the Customer, part of a Standard Taxonomy. • Valid values available in API version 62.0 and later are: – `ADG` – `Calculated_Insight` – `CG_Audience` – `Chunk` – `Directory_Table` – `External` – `Semantic` – `Transform` – `Vector_Embedding`
`datatype`	string	Required. Phone, currency, number, or other assigned type.
`dateFormat`	string	Optional: The Date format of date, time, date/time fields in this Transport field.
`externalName`	string	Optional. Name of the object in the external system (different from Developer Name).
`isDataRequired`	boolean	Optional. If true, data is required for this field.
`length`	int	Optional. Length of a string column
`masterLabel`	string	Optional? Field label.
`precision`	int	Optional. Used for currency and numeric accuracy.
`primaryIndexOrder`	int	Optional. If supplied, indicates this field is part of the primary key where the number value (starting at 1) indicates the order of attributes if this happens to be a compound primary key. Missing value means this field is not part of the primary key.
`scale`	int	Optional. Used for currency and numeric accuracy.
`sequence`	int	Optional. The sequence of this source schema.

ObjectSourceTargetMap

Contains the object-level mappings between the source and the target objects. The source and target objects can be an MktDataLakeObject or an MktDataModelObject. For example, an Email source object can be mapped to the ContactPointEmail object.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

ObjectSourceTargetMap components have the suffix `.objectSourceTargetMap` and are stored in the `objectSourceTargetMaps` folder.

Version

ObjectSourceTargetMap components are available in API version 51.0 and later.

Special Access Rules

To access this metadata type, you must have the Customize Application user permission.

Fields

Field Name	Field Type	Description
`creationType`	DefinitionCreationType (enumeration of type string)	Describes whether this object was added by the user or as part of a standard taxonomy. Valid values are: • `ADG` • `Activation_Audience` • `Bridge` • `Calculated_Insight` • `CG_Audience` (Available in API version 62.0 and later) • `Chunk` • `Curated` • `Custom` • `Derived` • `Directory_Table` • `External` • `Ml_Prediction` • `Segment_Membership` • `Semantic` • `Standard` • `System` • `Transform` • `Vector_Embedding`
`fieldSourceTargetMaps`	FieldSourceTargetMap[]	Contains the field-level mappings associated with this object mapping.
`masterLabel`	string	Required. The UI name for the target map.
`sequenceNbr`	int	Use this field to display multiple mappings between the same two objects, for a consistent customer experience when presenting the mappings.

Field Name	Field Type	Description
`sourceObjectName`	string	Required. Name of the source object that's mapped, such as Email, or SfmcEnt1_Subscriber.
`targetObjectName`	string	Required. Name of the target object that's mapped, such as ContactPointEmail or Individual.

FieldSourceTargetMap

Contains the field-level mappings between the source and the target objects.

The source and target can be MktDataLakeField or MktDataModelField.

For example, you can map a Person source object's field called emailAddress to an Individual target object's field called emailAddress.

Field Name	Field Type	Description
`creationType`	DefinitionCreationType (enumeration of type string)	Describes whether this object was added by the user or as part of a standard taxonomy. Values are: • `ADG` • `Activation_Audience` • `Bridge` • `Calculated_Insight` • `Chunk` • `Curated` • `Custom` • `Derived` • `Directory_Table` • `External` • `Ml_Prediction` • `Segment_Membership` • `Semantic` • `Standard` • `System` • `Transform` • `Vector_Embedding`
`filterApplied`	boolean	Specifies whether the field-level mapping is an event type filter (`true`) or not (`false`).
`filterOperationType`	string	If the field-level mapping is an event type filter, specifies the filtering operator. Value is: • `Equal`

Field Name	Field Type	Description
`filterValue`	string	If the field-level mapping is an event type filter, specifies the object that contains the event type field.
`isSourceFormula`	boolean	Specifies whether the source field is a formula (`true`) or not (`false`). If `true`, you must include the sourceFormula value.
`sourceField`	string	Required. The source object field that's mapped, such as `EmailAddr` or `SfmcEnt1_Subscriber.FName`.
`sourceFormula`	string	A formula, such as concatenation, date function, or constant value.
`targetField`	string	Required. The target object field that's mapped, such as `SfmcEnt1_Email.EmailAddr` or `Individual.FirstName`.

StreamingAppDataConnector

Represents the connection information specific to Web and Mobile Connectors.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

StreamingAppDataConnector components have the suffix `.streamingAppDataConnector` and are stored in the `streamingAppDataConnectors` folder.

Version

StreamingAppDataConnector components are available in API version 55.0 and later.

Special Access Rules

There are no additional access requirements that are specific to this type.

Fields

Field Name	Description
`appIdentifier`	**Field Type** string **Description** Required.

Field Name	Description
	The unique app identifier (UUID).
`dataConnectorType`	**Field Type** DataConnectorType (enumeration of type string) **Description** Required. The value of the field is restricted to `SteamingApp`. Possible values are: • `DataCloud` • `StreamingApp`
`isProtected`	**Field Type** boolean **Description** An auto-generated value that doesn't impact the behavior of the metadata type.
`masterLabel`	**Field Type** string **Description** Required. The display name of the connector.
`streamingAppDataConnectorType`	**Field Type** StreamingAppDataConnectorType (enumeration of type string) **Description** Required. The type of connector. Possible values are: • `MobileApp` • `WebApp`

Declarative Metadata Sample Definition

The following is an example of a StreamingAppDataConnector component.

```
<?xml version="1.0" encoding="UTF-8"?>
<StreamingAppDataConnector xmlns="http://soap.sforce.com/2006/04/metadata">
    <appIdentifier>61826b62-6b90-49ff-8259</appIdentifier>
    <dataConnectorType>StreamingApp</dataConnectorType>
    <masterLabel>My Web Application</masterLabel>
```

```
    <streamingAppDataConnectorType>WebApp</streamingAppDataConnectorType>
</StreamingAppDataConnector>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <fullName>MyPackage</fullName>
    <namespacePrefix>ns1</namespacePrefix>
    <types>
        <members>My_Web_Application_Behavioral_Events_F4DA8759</members>
        <name>DataStreamDefinition</name>
    </types>
    <types>
        <members>My_Web_Application_61826b62_6b90_49ff_8259</members>
        <name>ExternalDataConnector</name>
    </types>
    <types>
        <members>My_Web_Application_61826b62_6b90_49ff_8259</members>
        <name>StreamingAppDataConnector</name>
    </types>
    <version>55.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

AccountRelationshipShareRule

The rule that determines which object records are shared, how they're shared, the account relationship type that shares the records, and the level of access granted to the records.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

This type extends the MetadataWithContent metadata type and inherits its `content` and `fullName` fields.

File Suffix and Directory Location

AccountRelationshipShareRule components have the suffix `.accountRelationshipShareRule` and are stored in the `.accountRelationshipShareRules` folder.

Version

AccountRelationshipShareRule components are available in API version 45.0 and later.

Special Access Rules

Access to the AccountRelationshipShareRule type requires orgs to enable the Account Relationships permission. The Manage Experiences permission is required for user access.

Fields

Field Name	Field Type	Description
`accessLevel`	string	Type of access granted by the share rule. Valid values are: • Read • Edit
`accountToCriteriaField`	string	Criteria that must be met for the data to be shared. Valid values include any custom or standard lookup to Account or User on top-level objects. To get the full list for your org, do a Describe on the ARSR entity.
`description`	string	A meaningful explanation of the sharing rule.
`entityType`	string	The type of data shared by this share rule. Valid values are: • Account • Campaign • Case • Contact • Custom Object • Lead • Opportunity • Order API names of top-level custom objects in the org can also be used, for example, CustomObject__c.
`masterLabel`	string	The label assigned to the sharing rule to identify it.
`staticFormulaCriteria`	string	A way to further filter what data gets shared. This string must be a deterministic formula, and spanning isn't allowed.
`type`	string	Match the type of an account relationship for data to be shared according to the accountToCriteriaField and the staticFormulaCriteria fields.

Declarative Metadata Sample Definition

The following is an example of an AccountRelationshipShareRule component.

```
<?xml version="1.0" encoding="UTF-8"?>
<AccountRelationshipShareRule xmlns="http://soap.sforce.com/2006/04/metadata">
    <accessLevel>Edit</accessLevel>
    <accountToCriteriaField>Account.OwnerId</accountToCriteriaField>
```

```
    <description>TestDescription</description>
    <entityType>Account</entityType>
    <masterLabel>TestName</masterLabel>
    <staticFormulaCriteria>YearStarted = &quot;1980&quot;</staticFormulaCriteria>
    <type>Dealer</type>
</AccountRelationshipShareRule>
```

The following is an example `package.xml` that references the previous definition.

```
<Package>
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>ArsrDevName</members>
        <name>AccountRelationshipShareRule</name>
    </types>
<version>45.0</version>
</Package>
```

This metadata type supports the wildcard character * (asterisk) in the package.xml manifest file.

AccountingFieldMapping

Represents the accounting field mappings to organize your data and bring it to ledger entry records.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

AccountingFieldMapping components have the suffix `.accountingFieldMapping` and are stored in the `accountingFieldMappings` folder.

Version

AccountingFieldMapping components are available in API version 58.0 and later.

Fields

Field Name	Description
`accountingModelConfig`	**Field Type** string **Description** Required.

Field Name	Description
	Record ID of the AccountingModelConfig record that the Field Mapping is associated with.
`isForAllocationType`	**Field Type** boolean **Description** Reserved for internal use.
`isForPaymentType`	**Field Type** boolean **Description** Reserved for internal use.
`isForTransactionType`	**Field Type** boolean **Description** Reserved for internal use.
`isProtected`	**Field Type** boolean **Description** Indicates whether this component is protected (`true`) or not protected (`false`). Default value is `false`.
`mappingBehavior`	**Field Type** MappingBehaviorType (enumeration of type string) **Description** Required. Specifies how the target's field data is mapped from the source field only when the journal entry is created. When set to `CurrentValue`, Subledger reverses and replaces journal entries whose value differs from the value in `sourceField`. Valid values are: • `CurrentValue` • `PointInTime`
`masterLabel`	**Field Type** string **Description** Required. A user-friendly name for AccountingFieldMapping, which is defined when the AccountingFieldMapping is created.

Field Name	Description
sourceField	**Field Type** string **Description** The API name of the field on the source object that is mapped to the target field.
targetField	**Field Type** string **Description** Required. The API name of the field on the Transaction Journal record for this mapping.

Declarative Metadata Sample Definition

The following is an example of an AccountingFieldMapping component.

```
<?xml version="1.0" encoding="UTF-8"?>
<AccountingFieldMapping xmlns="http://soap.sforce.com/2006/04/metadata">
 <accountingModelConfig>ModelConfigOne</accountingModelConfig>
 <fullName>FieldMappingOne</fullName>
 <masterLabel>FieldMappingOne</masterLabel>
 <isForAllocationType>true</isForAllocationType>
 <isForPaymentType>true</isForPaymentType>
 <isForTransactionType>true</isForTransactionType>
 <mappingBehavior>PointInTime</mappingBehavior>
 <sourceField>TransactionJournal.MappingTargetOne__c</sourceField>
 <targetField>MappingTargetOne__c</targetField>
 <isProtected>false</isProtected>
</AccountingFieldMapping>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package
 xmlns="http://soap.sforce.com/2006/04/metadata">
 <types>
  <members>FieldMappingOne</members>
  <name>AccountingFieldMapping</name>
 </types>
 <version>58.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

AccountingModelConfig

Represents the mapping of the financial data model to a logical data model and configuration for the generation of Transaction Journal records.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

AccountingModelConfig components have the suffix `.accountingModelConfig` and are stored in the `accountingModelConfigs` folder.

Version

AccountingModelConfig components are available in API version 57.0 and later.

Fields

Field Name	Description
`accountingType`	**Field Type** AccountingType (enumeration of type string) **Description** Required. Determines whether the accounting set generates revenue or expense type transaction journal records. Valid values are: • `Expense` • `Revenue`
`defaultAccrualAccountCode`	**Field Type** string **Description** The code for your accounting system's default accrual account.
`defaultWriteOffAccountCode`	**Field Type** string **Description** Represents the name of your account for written off payments.

Field Name	Description
`earliestCreatedDate`	**Field Type** dateTime **Description** Required. The date used to filter source records for processing. The Accounting Subledger only considers records created on or after this date.
`expectedCashFlowGrouping`	**Field Type** ExpectedCashFlowGrouping (enumeration of type string) **Description** Determines whether Accounting Subledger groups transaction journal records by fund account or by a combination of fund account and due date. Note: Changing this setting doesn't impact existing records; it only affects records created or reversed afterward. Valid values are: • `GroupByFundAccount` • `GroupByFundAndDueDate`
`financeBook`	**Field Type** string **Description** Reserved for internal use.
`internalMappingDetails`	**Field Type** string **Description** Required. Represents the structure of your financial data in JSON format.
`isActive`	**Field Type** boolean **Description** Required. Indicates whether only records that are true are processed when the Subledger Job runs.
`isGroupedByFundAccount`	**Field Type** boolean **Description** Reserved for internal use.

Field Name	Description
`isUsed`	**Field Type** boolean **Description** Required. Indicates whether the Accounting Model has been used or activated at least once (`true`) or not (`false`). Note: If the value is set to `true`, you can't select another object for the object model or change the number of objects associated with that Accounting Model.
`jobFilterCriteria`	**Field Type** string **Description** Reserved for internal use.
`masterLabel`	**Field Type** string **Description** Required. A user-friendly name for AccountingModelConfig, which is defined when the AccountingModelConfig is created.
`paidCashFlowGrouping`	**Field Type** PaidCashFlowGrouping (enumeration of type string) **Description** Determines the level of detail for generated transaction journal records. Valid values are: • `GroupByFundAccount`—Accounting Subledger splits all transaction journal records into fund accounts. Secondary records are created for payment type records but not for transaction type records. • `GroupBySummary`—Accounting Subledger only splits credits for revenue and debits for expenses by fund accounts.
`recordTypeFilter`	**Field Type** string **Description** Specify the record type IDs from the primary object to be processed. This field is case-sensitive. Note: If no record type is specified in the filter, all records are processed.

Field Name	Description
runOrder	**Field Type** int **Description** Determines the load order sequence of the multiple Accounting Model. The lower number runs first. For example, load order 1 runs before load order 2.

Declarative Metadata Sample Definition

The following is an example of an AccountingModelConfig component.

```
<?xml version="1.0" encoding="UTF-8"?>
<AccountingModelConfig
 xmlns="http://soap.sforce.com/2006/04/metadata">
 <fullName>ModelConfigOne</fullName>
 <masterLabel>ModelConfigOne</masterLabel>
 <defaultAccrualAccountCode>abc</defaultAccrualAccountCode>
 <defaultWriteOffAccountCode>abc</defaultWriteOffAccountCode>
 <isUsed>false</isUsed>
 <isActive>false</isActive>
 <runOrder>123</runOrder>
 <recordTypeFilter>abcabc</recordTypeFilter>
 <earliestCreatedDate>2021-12-01T00:00:00.000Z</earliestCreatedDate>
 <internalMappingDetails>abcabc</internalMappingDetails>
 <accountingType>Revenue</accountingType>
 <expectedCashFlowGrouping>GroupByFundAccount</expectedCashFlowGrouping>
 <paidCashFlowGrouping>GroupBySummary</paidCashFlowGrouping>
</AccountingModelConfig>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package
 xmlns="http://soap.sforce.com/2006/04/metadata">
 <types>
  <members>ModelConfigOne</members>
  <name>AccountingModelConfig</name>
 </types>
 <version>57.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ActionLinkGroupTemplate

Represents the action link group template. Action link templates let you reuse action link definitions and package and distribute action links. An action link is a button on a feed element. Clicking on an action link can take a user to another Web page, initiate a file download, or invoke an API call to an external server or Salesforce. Use action links to integrate Salesforce and third-party services into the feed. Every action link belongs to an action link group and action links within the group are mutually exclusive.

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

ActionLinkGroupTemplate components have the suffix `.actionLinkGroupTemplate` and are stored in the `actionLinkGroupTemplates` folder.

Version

ActionLinkGroupTemplate components are available in API version 33.0 and later.

Fields

Field Name	Field Type	Description
`actionLinkTemplates`	ActionLinkTemplate on page 237[]	Action link templates that are associated with the action link group template.
`category`	PlatformAction GroupCategory (enumeration of type string)	Required. The location of the action link group within the feed element. Values are: • `Primary`—The action link group is displayed in the body of the feed element. • `Overflow`—The action link group is displayed in the overflow menu of the feed element.
`executionsAllowed`	ActionLink ExecutionsAllowed (enumeration of type string)	Required. The number of times an action link can be executed. Values are: • `Once`—An action link can be executed only once across all users. • `OncePerUser`—An action link can be executed only once for each user. • `Unlimited`—An action link can be executed an unlimited number of times by each user. If the action link's `actionType` is `Api` or `ApiAsync`, you can't use this value.
`hoursUntilExpiration`	int	Required. The number of hours from when the action link group is created until it's removed from associated feed elements and can no longer be executed. The maximum value is 8,760.

Field Name	Field Type	Description
`isPublished`	boolean	Required. If `true`, the action link group template is published. Action link group templates shouldn't be published until at least one action link template is associated with it.
`name`	string	Required. The name of the action link group template to use in code.

ActionLinkTemplate

ActionLinkTemplate components are used to create multiple action links that share properties.

Field Name	Field Type	Description
`actionUrl`	string	Required. The action link URL. For example, a `Ui` action link URL is a Web page. A `Download` action link URL is a link to the file to download. `Ui` and `Download` action link URLs are provided to clients. An `Api` or `ApiAsync` action link URL is a REST resource. `Api` and `ApiAsync` action link URLs aren't provided to clients. Links to Salesforce can be relative. All other links must be absolute and start with `https://`.
`headers`	string	Template for the HTTP headers sent when corresponding action links are invoked. This field can be used only for `Api` and `ApiAsync` action links. This field can contain context variables and binding variables in the form `{!Bindings.`***`key`***`}`.
`isConfirmationRequired`	boolean	Required. If `true`, a confirmation dialog appears before the action is executed.
`isGroupDefault`	boolean	Required. If `true`, action links derived from this template are the default or primary action in their action groups. There can be only one default action per action group.
`label`	string	A custom label to display on the action link button. If none of the `LabelKey` values make sense for an action link, use a custom label. Set the `LabelKey` field to `None` and enter a label name in the `Label` field.
`labelKey`	string	Required. Key for the set of labels to display for these action link states: new, pending, success, failed. For example, the Approve set contains these labels: Approve, Pending, Approved, Failed. For a complete list of keys and labels, see Action Link Labels in the *Connect REST API Developer Guide*.
`linkType`	ActionLinkType (enumeration of type string)	Required. The type of action link. One of these values: • `Api`—The action link calls a synchronous API at the action URL. Salesforce sets the status to `SuccessfulStatus` or `FailedStatus` based on the HTTP status code returned by your server. • `ApiAsync`—The action link calls an asynchronous API at the action URL. The action remains in a `PendingStatus` state until a third party makes a request to `/connect/action-links/`***`actionLinkId`*** to set the status to `SuccessfulStatus` or `FailedStatus` when the asynchronous operation is complete. • `Download`—The action link downloads a file from the action URL.

Field Name	Field Type	Description
		• `Ui`—The action link takes the user to a web page at the action URL.
`method`	ActionLink HttpMethod (enumeration of type string)	Required. HTTP method for the action URL. One of these values: • `HttpDelete`—Returns HTTP 204 on success. Response body or output class is empty. • `HttpGet`—Returns HTTP 200 on success. • `HttpHead`—Returns HTTP 200 on success. Response body or output class is empty. • `HttpPatch`—Returns HTTP 200 on success or HTTP 204 if the response body or output class is empty. • `HttpPost`—Returns HTTP 201 on success or HTTP 204 if the response body or output class is empty. Exceptions are the batch posting resources and methods, which return HTTP 200 on success. • `HttpPut`—Return HTTP 200 on success or HTTP 204 if the response body or output class is empty. `Ui` and `Download` action links must use `HttpGet`.
`position`	int	Required. An integer specifying the position of the action link template relative to other action links in the group. 0 is the first position.
`requestBody`	string	Template for the HTTP request body sent when corresponding action links are invoked. This field can be used only for `Api` and `ApiAsync` action links. This field can contain context variables and binding variables in the form `{!Bindings.`***`key`***`}`.
`userAlias`	string	If you selected `CustomUser` or `CustomExcludedUser` for `UserVisibility`, this field is the alias for the custom user. Use the alias in a template binding to specify the custom user when an action link group is created using the template.
`userVisibility`	ActionLink UserVisibility (enumeration of type string)	Required. Who can see the action link. This value is set per action link, not per action link group. Values are: • `Creator`—Only the creator of the action link can see the action link. • `Everyone`—Everyone can see the action link. • `EveryoneButCreator`—Everyone but the creator of the action link can see the action link. • `Manager`—Only the manager of the creator of the action link can see the action link. • `CustomUser`—Only the custom user can see the action link. • `CustomExcludedUser`—Everyone but the custom user can see the action link.

Declarative Metadata Sample Definition

The following is an example of an ActionLinkGroupTemplate component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ActionLinkGroupTemplate xmlns="http://soap.sforce.com/2006/04/metadata">
   <actionLinkTemplates>
      <actionUrl>/services/data/{!Bindings.word}/chatter/feed-elements</actionUrl>
      <headers>Content-Type:{!Bindings.word3}</headers>
      <isConfirmationRequired>true</isConfirmationRequired>
      <isGroupDefault>true</isGroupDefault>
      <labelKey>Add</labelKey>
      <linkType>API</linkType>
      <method>httpPost</method>
      <position>0</position>
      <requestBody>{"body":{"messageSegments":[{"type": "Text",
      "text": "{!Bindings.word1}"}]},"subjectId": "{!Bindings.word2}",
      "feedElementType": "feedItem"}</requestBody>
      <userAlias>customExcludedUser</userAlias>
      <userVisibility>CustomExcludedUser</userVisibility>
   </actionLinkTemplates>
   <category>Primary</category>
   <executionsAllowed>OncePerUser</executionsAllowed>
   <hoursUntilExpiration>10</hoursUntilExpiration>
   <isPublished>true</isPublished>
   <name>MyPackage</name>
</ActionLinkGroupTemplate>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>ActionLinkGroupTemplate</name>
    </types>
    <version>33.0</version>
</Package>
```

Usage

If you modify action link group templates, you overwrite the related action link templates.

If you delete a published action link group template, you delete all related action link information which includes deleting all action links that were instantiated using the template from feed items.

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ActionPlanTemplate

Represents the instance of an action plan template.This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

ActionPlanTemplate components have the suffix `.apt` and are stored in the `actionPlanTemplates` folder.

Version

Action Plan Template components are available in API version 47.0 and later.

Special Access Rules

To create or access action plan templates, you must have the Customize Application permission and the IndustriesActionPlans license.

Fields

Field Name	Field Type	Description
`actionPlanTemplateItem`	ActionPlanTemplateItem on page 240	The instance of an item on an action plan template version.
`description`	string	The description of the action plan template.
`name`	string	Required. The name of the action plan template.
`targetEntityType`	string	Required. The parent object this action plan template relates to. Supported parent objects are Account, BusinessMilestone, Campaign, Case, Claim, Contact, Contract, InsurancePolicy, InsurancePolicyCoverage, Lead, Opportunity, PersonLifeEvent, and Visit and custom objects with activities enabled.
`uniqueName`	string	Required. The unique identifier for this action plan template record.

ActionPlanTemplateItem

Represents the instance of an item on an action plan template version.

Field Name	Field Type	Description
`actionPlanTemplateItemValue`	actionPlanTemplateItemValue	The value associated with the action plan template item.
`displayOrder`	int	The order in which this item is displayed within the action plan template version.
`isRequired`	boolean	Indicates whether the task created from this template item is required.
`itemEntityType`	string	Required. The name of the field on the action plan template item that this value is for. Available in API version 48.0 and later.

Field Name	Field Type	Description
name	string	Required. The name of the action plan template item.
uniqueName	string	Required. The unique identifier for this action plan template item record.

ActionPlanTemplateItemValue

Represents the value associated with an action plan template item.

Field Name	Field Type	Description
itemEntityType	string	Required. The name of the field on the action plan template item that this value is for. Available in API version 48.0 and later.
name	string	Required. The name of the action plan template item value.
valueFormula	string	The formula for this action plan template item.
valueLiteral	string	The value for this action plan template item.

Declarative Metadata Sample Definition

The following is an example of an ActionPlanTemplate component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ActionPlanTemplate xmlns="http://soap.sforce.com/2006/04/metadata">
    <actionPlanTemplateItem>
        <actionPlanTemplateItemValue>
            <itemEntityType>Task</itemEntityType>
            <name>Subject</name>
            <valueLiteral>Task 1</valueLiteral>
        </actionPlanTemplateItemValue>
        <actionPlanTemplateItemValue>
            <itemEntityType>Task</itemEntityType>
            <name>Priority</name>
            <valueLiteral>Normal</valueLiteral>
        </actionPlanTemplateItemValue>
        <actionPlanTemplateItemValue>
            <itemEntityType>Task</itemEntityType>
            <name>ActivityDate</name>
            <valueFormula>StartDate + 1</valueFormula>
        </actionPlanTemplateItemValue>
        <isRequired>true</isRequired>
        <itemEntityType>Task</itemEntityType>
        <name>Task 1</name>
        <uniqueName>Task_1_f1beaba5_1ae1_11ea_93ad_5bc214d4daaf</uniqueName>
    </actionPlanTemplateItem>
    <description>This is a sample template for packaging</description>
    <name>Action Plan Template 1</name>
    <targetEntityType>Account</targetEntityType>
    <uniqueName>Action_Plan_Template_1_da365953_1ae1_11ea_93ad_89a1c3d51bae</uniqueName>
</ActionPlanTemplate>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ActionableListDefinition

Represents the data source definition details associated with an actionable list.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

ActionableListDefinition components have the suffix `.actionableListDefinition` and are stored in the `actionableListDefinitions` folder.

Version

ActionableListDefinition components are available in API version 57.0 and later.

Fields

Field Name	Description
`actionableListDatasetColumns`	**Field Type** ActionableListDatasetColumn[] **Description** The object that stores columns in a dataset associated with an actionable list.
`actionableListMemberStatuses`	**Field Type** ActionableListMemberStatus[] **Description** The object that stores the status and the corresponding status icon details of an individual actionable list member.
`batchCalcJobDefinition`	**Field Type** string **Description** The batch calculation job definition that's associated with the creation of an actionable list. This field is a relationship field.

Field Name	Description
`datasetName`	**Field Type** string **Description** The name of the dataset that is associated with the actionable list.
`edgeMart`	**Field Type** string **Description** The edgemart dataset that's associated with the actionable list. Available in API version 58.0 and later.
`isActive`	**Field Type** boolean **Description** Indicates whether the actionable list definition is active (`true`) or not (`false`). The default value is `false`.
`masterLabel`	**Field Type** string **Description** Required. The master label of the actionable list definition.
`objectName`	**Field Type** string **Description** Required. The object for which the actionable list is created. Possible values are organized by the API version in which they were introduced. Values are available in all versions after introduction unless noted otherwise. Possible values are: API version 60.0 and later: `PersonLifeEvent` API version 60.0 and later with Insurance Managed Package: • `Claim` • `InsurancePolicy` • `Quote` API version 59.0 and later with Health Cloud: • `CareFacilityBed`

Field Name	Description
	• `CareRequest` • `CareRequestItem` • `CareServiceVisit` • `CareServiceVisitPlan` • `ClinicalServiceRequest` API version 59.0 and later with Loyalty Cloud: • `LoyaltyProgramMember` API version 59.0 and later: • `Case` API version 58.0 and later with Automotive Cloud: • `Vehicle` API version 58.0 and later: • `Asset` • `Lead` • `Opportunity` API version 57.0 and later: • `Account` • `Contact`

ActionableListDatasetColumn

Represents the information about the columns in a dataset associated with an actionable list.

Table 1: Fields

Field Name	Description
`dataDomain`	**Field Type** DatasetColumnDataType (enumeration of type string) **Description** The data domain that is mapped to the data type of the dataset column. Possible values are: • `Dates` • `Dimensions` • `Measures`
`dataType`	**Field Type** DatatableDataType (enumeration of type string) **Properties** Create, Filter, Group, Nillable, Restricted picklist, Sort, Update

Field Name	Description
	Description The data type of the dataset column in the actionable list. Available in API version 58.0 and later. Possible values are: • `Boolean` • `Currency` • `Date` • `DateTime` • `Email` • `Location` • `Number` • `Percent` • `Phone` • `Text` • `Url`
`displayOrder`	**Field Type** int **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** The order in which the actionable list dataset columns are displayed. Available in API version 58.0 and later.
`isDefault`	**Field Type** boolean **Description** Indicates whether the dataset column is added to the actionable list by default (`true`) or not (`false`). The default value is `false`.
`isGroupedByListDefObj`	**Field Type** boolean **Description** Indicates whether the dataset column is grouped by the object defined in the actionable list definition (`true`) or not (`false`). Available in API version 59.0 and later.
`IsTypeAheadSearchEnabled`	**Field Type** boolean

Field Name	Description
	Description Indicates whether the type-ahead search for filters is enabled (`true`) or not (`false`). Available in API version 60.0 and later.
`objectName`	**Field Type** string **Description** The name of the object that's associated with the dataset column. Possible values are: API version 60.0 and later: `PersonLifeEvent` API version 60.0 and later with Insurance Managed Package: • `Claim` • `InsurancePolicy` • `Quote` API version 59.0 and later with Health Cloud: • `CareFacilityBed` • `CareRequest` • `CareRequestItem` • `CareServiceVisit` • `CareServiceVisitPlan` • `ClinicalServiceRequest` API version 59.0 and later with Loyalty Cloud: • `LoyaltyProgramMember` API version 59.0 and later: • `Case` API version 58.0 and later with Automotive Cloud: • `Vehicle` API version 58.0 and later: • `Asset` • `Lead` • `Opportunity` API version 57.0 and later: • `Account` • `Contact`
`sourceColumnApiName`	**Field Type** string

Field Name	Description
	Description The API name of the column in the source dataset.
sourceFieldName	**Field Type** string **Description** The name of the field in the object for which the actionable list dataset is created.

ActionableListMemberStatus

Represents the status and the corresponding status icon details of an individual actionable list member.

Table 2: Fields

Field Name	Description
iconName	**Field Type** string **Description** The name of the icon that's mapped to the status.
status	**Field Type** string **Description** The status of the actionable list member.

Declarative Metadata Sample Definition

The following is an example of a ActionableListDefinition component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ActionableListDefinition
 xmlns="http://soap.sforce.com/2006/04/metadata">
 <actionableListDatasetColumns>
  <isDefault>true</isDefault>
  <sourceFieldName>NewColumn1</sourceFieldName>
 </actionableListDatasetColumns>
 <actionableListDatasetColumns>
  <sourceColumnApiName>ApiName</sourceColumnApiName>
  <dataDomain>Dimensions</dataDomain>
  <isDefault>false</isDefault>
  <sourceFieldName>NewColumn2</sourceFieldName>
  <objectName>Account</objectName>
  <displayOrder>1</displayOrder>
  <dataType>Text</dataType>
```

```
 </actionableListDatasetColumns>
 <actionableListMemberStatuses>
  <iconName>NewMember1</iconName>
  <status>Active</status>
 </actionableListMemberStatuses>
 <isActive>true</isActive>
 <masterLabel>NewMember2</masterLabel>
 <objectName>Account</objectName>
 <isProtected>true</isProtected>
 <batchCalcJobDefinition>Test1</batchCalcJobDefinition>
 <datasetName>AccountDef</datasetName>
</ActionableListDefinition>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
   <types>
      <members>*</members>
      <name>ActionableListDefinition</name>
   </types>
   <version>64.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

AdvAccountForecastSet

Represents the forecast sets that define the forecast configurations for each business unit or different groups of accounts. With separate forecast sets at account or business unit level, you can focus on account-specific data and manage configuration updates for one business unit without impacting any other business unit's data.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

AdvAccountForecastSet components have the suffix `.advAccountForecastSet` and are stored in the `AdvAccountForecastSet` folder.

Version

AdvAccountForecastSet components are available in API version 53.0 and later.

Special Access Rules

The advanced account forecasting feature setting for Manufacturing Cloud is required to create an advanced account forecast set.

Fields

Field Name	Description
`accountFieldName`	**Field Type** string **Description** The field name for the account in the advanced account forecast fact record.
`calculationFrequency`	**Field Type** AdvAcctFcstCalcFrequency (enumeration of type string) **Description** The frequency at which the forecast set is recalculated automatically. Possible values are: • `Monthly` • `Quarterly` • `Weekly` • `Yearly` The default value is `Monthly`.
`description`	**Field Type** string **Description** The description of the advanced account forecast set record.
`dimensions`	**Field Type** AdvAcctForecastDimension[] **Description** The dimensions selected for an advanced account forecast set to categorize the forecast data.
`displayGroups`	**Field Type** AdvAcctFrcstDisplayGroup[] **Description** The information about the groups for the advanced account forecast set measures or dimensions.
`forecastAdjPeriods`	**Field Type** AdvAcctForecastAdjPeriod[]

Field Name	Description
	Description The details about the adjustment period of the advanced account forecast values.
`forecastFactObjectName`	**Field Type** string **Description** Required. The API name of the object that contains the advanced forecast fact records.
`forecastFormulas`	**Field Type** AdvAccountForecastFormula[] **Description** The formulas based on which forecast values are calculated.
`forecastPeriodGroupName`	**Field Type** string **Description** Required. The name of the advanced account forecast period group record.
`forecastQuantityFieldName`	**Field Type** string **Description** The field name for the forecast quantity in the advanced account forecast fact record.
`forecastRevenueFieldName`	**Field Type** string **Description** The field name for the forecast revenue in the advanced account forecast record.
`forecastSetFieldName`	**Field Type** string **Description** The field name for the Forecast Set ID in the advanced account forecast fact record.
`forecastSetName`	**Field Type** string **Description** Required. The name of the advanced account forecast set record.

Field Name	Description
`forecastStatusFieldName`	**Field Type** string **Description** The field name for the status in the advanced account forecast fact record.
`generationDpeDefName`	**Field Type** string **Description** The name of the data processing engine (DPE) definition that's used to generate advanced account forecast fact records.
`measureDefinitions`	**Field Type** AdvAcctForecastMeasureDef[] **Description** The measures to display in the advanced account forecasts grid for the forecast set.
`periodFieldName`	**Field Type** string **Description** The field name for the period in the advanced account forecast fact record.
`recalculateDpeDefName`	**Field Type** string **Description** The name of the data processing engine definition that's used to recalculate the advanced account forecast fact records.
`regenerationDpeDefName`	**Field Type** string **Description** The name of the data processing engine definition that's used to regenerate the advanced account forecast fact records.
`rolloverDpeDefName`	**Field Type** string **Description** The data processing engine definition that's used to generate the rollover advanced account forecast fact records.
`rolloverFrequency`	**Field Type** AdvAcctFcstCalcFrequency (enumeration of type string)

Field Name	Description
	Description The frequency of rollover of the advanced account forecast records. Possible values are: • `Monthly` • `Quarterly` • `Weekly` • `Yearly` The default value is `Monthly`.
`status`	**Field Type** AdvAccForecastSetStatus (enumeration of type string) **Description** Required. The status of the advanced account forecast set. Possible values are: • `Active` • `Inactive`

AdvAccountForecastFormula

Represents the formulas that are used to calculate forecast values in real time after applying the DPE calculations. For example, processing forecast rollover for all accounts at the start of a month.

Field Name	Description
`endPeriod`	**Field Type** int **Description** Required. The period until when the forecast formula is applicable.
`formulaExpression`	**Field Type** string **Description** Required. The formula based on which forecast values are calculated.
`formulaType`	**Field Type** AdvAcctFcstFormulaType (enumeration of type string)

Field Name	Description
	Description Required. Specifies the calculation type for the formula. Possible values are: • `QUANTITY` • `REVENUE` The default value is `QUANTITY`.
`startPeriod`	**Field Type** int **Description** Required. The period from when the forecast formula is applicable.

AdvAcctForecastAdjPeriod

Represents details about the adjustment period of the advanced account forecast values.

Field Name	Description
`adjustmentDayCount`	**Field Type** int **Description** Required. The number of days during which you can make forecast adjustments.
`frequency`	**Field Type** PeriodTypes (enumeration of type string) **Description** Required. The frequency that's applicable to make any forecast adjustments. Possible values are: • `Month` • `Quarter` • `Week` • `Year` The default value is `Month`.

Field Name	Description
profileName	**Field Type** string **Description** The name of the profile with which you can adjust the forecast set.
startDay	**Field Type** int **Description** Required. The start date for forecast adjustments.

AdvAcctForecastDimension

Represents the dimensions selected for an advanced account forecast set to categorize the data. For example, a business unit requires forecast data for each account aggregated by product and ship-from location.

Field Name	Description
advAcctForecastDimName	**Field Type** string **Description** Required. The name of the advanced account forecast dimension.
dimensionFieldName	**Field Type** string **Description** Required. The API name of the field for the dimension in the custom object that contains the generated advanced account forecast records.
dimensionSourceName	**Field Type** string **Description** The name of the dimension source associated with the advanced account forecast set dimension record.
hierarchySequenceNumber	**Field Type** int

Field Name	Description
	Description Required. The sequence number of the dimension source associated with the forecast set.

AdvAcctForecastMeasureDef

Represents information about the measures to display in the advanced account forecasts grid for the forecast set.

Field Name	Description
`advAcctForecastMeasureDefName`	**Field Type** string **Description** Required. The name of the definition for the advanced account forecast measure.
`aggregationType`	**Field Type** AdvAcctFcstAggregationType (enumeration of type string) **Description** Required. The type of aggregation that's used for calculating the advanced account forecast values. Possible values are: • `AVERAGE` • `MAXIMUM` • `MINIMUM` • `SUM` The default value is `SUM`.
`computationMethod`	**Field Type** AdvAcctFcstComputationMethod (enumeration of type string) **Description** Required. The method that's used for calculating the advanced account forecast values. Values are: • `CUSTOM` • `DATA_PROCESSING_ENGINE_DEFINITION` • `FORMULA`

Field Name	Description
	The default value is `DATA_PROCESSING_ENGINE_DEFINITION`.
`forecastDataMeasureName`	**Field Type** string **Description** Required. The field of the facts object used for the advanced account forecast measure.
`forecastMeasureName`	**Field Type** string **Description** Required. The name for the advanced account forecast measure to show on UI.
`forecastMeasureType`	**Field Type** AdvAcctFcstMeasureType (enumeration of type string) **Description** Required. The measure type that's used for the generated advanced forecast values. Possible values are: • `QUANTITY` • `REVENUE` The default value is `QUANTITY`.
`isAdjustmentTracked`	**Field Type** boolean **Description** Indicates whether the adjustments made to the advanced account forecast values for this metric are tracked (`true`) or not (`false`).

AdvAcctFrcstDisplayGroup

Represents information about the groups for the advanced account forecast set measures or dimensions.

Field Name	Description
`advAcctFrcstDisplayGroupName`	**Field Type** string **Description** Required.

Field Name	Description
	The name of the advanced account forecast display group.
`displayGroupItems`	**Field Type** AdvAcctFrcstDplyGroupItem[] **Description** The items associated with a display group for an advanced account forecast set.
`displayGroupType`	**Field Type** AdvAcctFrcstDisplayGroupType (enumeration of type string) **Description** The category of the display group. Possible value is: • `MEASURE`
`isDefault`	**Field Type** boolean **Description** Indicates whether the display group is the default group (`true`) or not (`false`).
`userProfileName`	**Field Type** string **Description** The name of the profile for which the display group is applicable.

AdvAcctFrcstDplyGroupItem

Represents information about the items associated with a display group for an advanced account forecast set.

Field Name	Description
`advAcctFrcstDplyGroupItemName`	**Field Type** string **Description** Required. The name of the advanced account forecast display group that's associated with the group item.
`displayOrder`	**Field Type** int **Description** Required.

Field Name	Description
	The display order of the display group item.
measureReferenceName	**Field Type** string **Description** The ID of the measure associated with the display group item.

Declarative Metadata Sample Definition

The following is an example of an AdvAccountForecastSet component.

```
<?xml version="1.0" encoding="UTF-8"?>
<AdvAccountForecastSet xmlns="http://soap.sforce.com/2006/04/metadata"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <calculationFrequency>Quarterly</calculationFrequency>
    <forecastAdjPeriods>
        <adjustmentDayCount>5</adjustmentDayCount>
        <frequency>Quarter</frequency>
        <profileName xsi:nil="true"/>
        <startDay>1</startDay>
    </forecastAdjPeriods>
    <forecastFormulas>
        <endPeriod>12</endPeriod>
        <formulaExpression>6</formulaExpression>
        <startPeriod>2</startPeriod>
        <formulaType>QUANTITY</formulaType>
    </forecastFormulas>
    <forecastPeriodGroupName>PeriodGroup1</forecastPeriodGroupName>
    <accountFieldName>Account</accountFieldName>
    <periodFieldName>Period</periodFieldName>
    <forecastQuantityFieldName>ForecastedQuantity</forecastQuantityFieldName>
    <forecastRevenueFieldName>ForecastedRevenue</forecastRevenueFieldName>
    <forecastFactObjectName>AdvAccountForecastFact</forecastFactObjectName>
    <forecastSetFieldName>AdvAcctForecastSetPartner</forecastSetFieldName>
    <rolloverFrequency>Monthly</rolloverFrequency>
    <forecastStatusFieldName>Status</forecastStatusFieldName>
    <description>sample forecast set</description>
    <regenerationDpeDefName xsi:nil="true"/>
    <rolloverDpeDefName xsi:nil="true"/>
    <recalculateDpeDefName xsi:nil="true"/>
    <generationDpeDefName xsi:nil="true"/>
    <status>Inactive</status>
    <forecastSetName>Forecast Set 1</forecastSetName>
    <dimensions>
        <dimensionFieldName>Account</dimensionFieldName>
        <dimensionSourceName>DimSource1</dimensionSourceName>
        <hierarchySequenceNumber>1</hierarchySequenceNumber>
        <advAcctForecastDimName>DimensionName</advAcctForecastDimName>
    </dimensions>
    <measureDefinitions>
```

```
        <forecastDataMeasureName>MeasureName</forecastDataMeasureName>
        <advAcctForecastMeasureDefName>Sample Def Name</advAcctForecastMeasureDefName>
        <isAdjustmentTracked>true</isAdjustmentTracked>
        <forecastMeasureName>Samplemeasure name</forecastMeasureName>
        <aggregationType>MINIMUM</aggregationType>
        <computationMethod>DATA_PROCESSING_ENGINE_DEFINITION</computationMethod>
        <forecastMeasureType>QUANTITY</forecastMeasureType>
    </measureDefinitions>
    <displayGroups>
        <advAcctFrcstDisplayGroupName>Sample Measure Group</advAcctFrcstDisplayGroupName>

        <displayGroupType>MEASURE</displayGroupType>
        <isDefault>false</isDefault>
        <userProfileName xsi:nil="true"/>
        <displayGroupItems>
           <advAcctFrcstDplyGroupItemName>Sample Quantity</advAcctFrcstDplyGroupItemName>

            <measureReferenceName>Sample Def Name</measureReferenceName>
            <displayOrder>1</displayOrder>
        </displayGroupItems>
    </displayGroups>
</AdvAccountForecastSet>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
       <members>*</members>
       <name>AdvAccountForecastSet</name>
    </types>
   <version>64.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

AffinityScoreDefinition

Represents the affinity information used in calculations to analyze and categorize contacts for marketing purposes.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

AffinityScoreDefinition components have the suffix `.affinityScoreDefinition` and are stored in the `affinityScoreDefinitions` folder.

Version

AffinityScoreDefinition components are available in API version 64.0 and later.

Special Access Rules

This metadata type is available only if the Fundraising Access license is enabled for the org and the Fundraising admin permission is assigned to users.

Fields

Field Name	Description
`affinityScoreDefinitionDesc`	**Field Type** string **Description** Description of the affinity score definition.
`affinityScoreDefinitionName`	**Field Type** string **Description** Name of the affinity score definition.
`affinityScoreType`	**Field Type** AffinityScoreType (enumeration of type string) **Description** Type of the affinity score that's defined. Valid values are: • `CAP`—Capacity, Ability, Propensity (CAP) • `RFM`—Recency, Frequency, Monetary (RFM) The default value is `RFM`.
`masterLabel`	**Field Type** string **Description** Label for this affinity score definition value. This display value is the internal label that doesn't get translated.

Field Name	Description
`numberOfMonths`	**Field Type** int **Description** Number of months to analyze the records for calculating the affinity score.
`numberOfRanges`	**Field Type** int **Description** Required. Number of ranges to use in the calculation, ranging from 0 to 9. Provide the corresponding range list values in the `scoreRangeList` field.
`scoreRangeList`	**Field Type** string **Description** Required. Ranges that are referenced in the affinity score calculation. This field is used with `scoreRangeList`. For example, to calculate RFM with `numberOfRanges` value as 3, provide the values for the `scoreRangeList` field in this format. `{ "R ranges":"0-30, 31-100, 100+", "F ranges":"0-10, 11-100, 100+", "M ranges":"0-1000, 1001-5000, 5000+" }`
`sourceFieldApiNameList`	**Field Type** string **Description** Required. API names of the source fields that are referenced in the score calculation.
`sourceObjectApiNameList`	**Field Type** string **Description** API names of the source objects that are referenced in the score calculation.
`targetFieldApiNameList`	**Field Type** string **Description** Required.

Field Name	Description
	API names of the target fields where the calculated scores are added.
`targetObjectApiName`	**Field Type** string API name of the target object where the calculated scores are added.

Declarative Metadata Sample Definition

This example shows a sample of an AffinityScoreDefinition component.

```
<?xml version="1.0" encoding="UTF-8"?>
<AffinityScoreDefinition
 xmlns="http://soap.sforce.com/2006/04/metadata">
 <affinityScoreDefinitionDesc>RFM Affinity Score</affinityScoreDefinitionDesc>
 <affinityScoreDefinitionName>AffinityScoreDefinition_RFM</affinityScoreDefinitionName>
 <affinityScoreType>RFM</affinityScoreType>
 <masterLabel>MasterLabel</masterLabel>
 <numberOfMonths>12</numberOfMonths>
 <numberOfRanges>3</numberOfRanges>
 <scoreRangeList>
        [
          {
            "name": "R Ranges",
            "direction": "ascending",
            "ranges": [30,90,180]
          },
          {
            "name": "F Ranges",
            "direction": "descending",
            "ranges": [10,15,100]
          },
          {
            "name": "M Ranges",
            "direction": "descending",
            "ranges": [500,1000,5000]
          }
     ]
      </scoreRangeList>
 <sourceFieldApiNameList>
        [
          {
            "name": "R Source",
            "values":
                [
                  {
                    "fieldName": "DonorGiftSummary.DaysSinceLastGift",
                    "fieldWeight": 1
                  }
                ]
          },
```

```
          {
            "name": "F Source",
            "values":
                [
                  {
                    "fieldName": "DonorGiftSummary.GiftCount",
                    "fieldWeight": 1
                  }
                ]
          },
          {
            "name": "M Source",
            "values":
                [
                  {
                    "fieldName": "DonorGiftSummary.TotalGiftsCount",
                    "fieldWeight": 1
                  }
                ]
          }
        ]
     </sourceFieldApiNameList>
<targetFieldApiNameList>
        [
          {
            "name": "R Target",
            "values":
                [
                  {
                    "fieldName": "DonorGiftSummary.RecencyScore",
                    "fieldWeight": 1
                  }
                ]
          },
          {
            "name": "F Target",
            "values":
                [
                  {
                    "fieldName": "DonorGiftSummary.FrequencyScore",
                    "fieldWeight": 1
                  }
                ]
          },
          {
            "name": "M Target",
            "values":
                [
                  {
                    "fieldName": "DonorGiftSummary.MonetaryScore",
                    "fieldWeight": 1
                  }
                ]
          }
```

```
        ]
      </targetFieldApiNameList>
</AffinityScoreDefinition>
```

This example shows a sample of the `package.xml` file that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>AffinityScoreDefinition</name>
    </types>
    <version>64.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

AIApplication

Represents an instance of an AI application. For example, Einstein Prediction Builder. This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

AIApplication components have the suffix `.ai` and are stored in the `aiApplications` folder.

Version

AIApplication is available in API version 50.0 and later.

Fields

Field Name	Field Type	Description
`developerName`	string	Required. Represents the name of the application. Can contain only underscores and alphanumeric characters and must be unique in your org. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. **Note:** Only users with View DeveloperName OR View Setup and Configuration permission can view, group, sort, and filter this field.
`masterLabel`	string	Label that identifies the AI application throughout the Salesforce user interface.

Field Name	Field Type	Description
`status`	AIApplicationStatus (enumeration of type string)	Required. The status of the application. Valid values are: • `Disabled` • `Draft` • `Enabled` • `Migrated`
`type`	AIApplicationType (enumeration of type string)	The type of AI application. Valid values are: • `PredictionBuilder`

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

AIApplicationConfig

Additional prediction information related to an AI application. This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

AIApplicationConfig components have the suffix `.aiapplicationconfig` and are stored in the `aiApplicationConfigs` folder.

Version

AIApplicationConfig is available in API version 50.0 and later.

Fields

Field Name	Field Type	Description
`aiApplicationDeveloperName`	string	Required. Represents the AIApplication to which AIApplicationConfig belongs. Can contain only underscores and alphanumeric characters and must be unique in your org. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. Available in API version 51.0 and later.
`applicationId`	string	Required. The ID of the parent AI application.
`developerName`	string	Represents the name of the application config. Can contain only underscores and alphanumeric characters and must be unique in your org. It must begin

Field Name	Field Type	Description
		with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
`insightReasonEnabled`	boolean	Required. When `true`, generates the predictors, or field values, that were used to generate the prediction value.
`masterLabel`	string	Required. Label that identifies the AI application configuration throughout the Salesforce user interface.
`rank`	int	Required. Reserved for future use.
`scoringMode`	AIScoringMode (enumeration of type string)	Required. Frequency with which the prediction scores are written back. Valid values are: • `Batch` • `OnDemand` • `Streaming`

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

AIScoringModelDefinition

Represents information about a machine learning model that's used by the Scoring Framework for Industries Cloud Einstein. The machine learning model is used for scoring, including its configuration.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

AIScoringModelDefinition components have the suffix `.aiScoringModelDefinition` and are stored in the `aiScoringModelDefinitions` folder.

Version

AIScoringModelDefinition components are available in API version 58.0 and later.

Special Access Rules

To access this metadata type, you must have the AI Accelerator User permission set with Scoring Framework enabled for Industries Cloud Einstein from Salesforce Setup. The Salesforce org must have the CRM Plus license and the product's CRM license.

Fields

Field Name	Description
`aiModelConfig`	**Field Type** string **Description** Required. ID of an AI model configuration related to the AI scoring model record.
`aiScoringModelDefVersions`	**Field Type** AIScoringModelDefVersion[] **Description** Represents information of various versions of a model.
`description`	**Field Type** string **Description** Description for an AIScoringModelDefinition record.
`masterLabel`	**Field Type** string **Description** Required. A user-friendly name for the AIScoringModelDefinition metadata component, which is defined when the AIScoringModelDefinition metadata component is created.

AIScoringModelDefVersion

Represents information about a version of an AI scoring model.

Field Name	Description
`aiScoringModelDefinition`	**Field Type** string **Description** Required.

Field Name	Description
	Parent AIScoringModelDefinition record that's related to an AIScoringModelDefVersion record.
`aiScoringSteps`	**Field Type** AIScoringStep[] **Description** Represents information about a step associated with an AI scoring model version.
`developerName`	**Field Type** string **Description** Required. The unique name of the object in the API. This name can contain only underscores and alphanumeric characters, and must be unique in your org. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. In managed packages, this field prevents naming conflicts on package installations. With this field, a developer can change the object's name in a managed package and the changes are reflected in a subscriber's organization. Label is **Record Type Name**.
`masterLabel`	**Field Type** string **Description** Required. A user-friendly name for the AIScoringModelDefVersion component name, which is defined when the AIScoringModelDefVersion component name is created.
`modelMode`	**Field Type** AIScoringModelDefVersionMode (enumeration of type string) **Description** Required. Mode of an AI scoring model. Values are: • `DEPLOY` • `TRAIN` • `TRAIN_AND_DEPLOY`

AIScoringStep

Represents information about a step associated with an AI scoring model version. For example, an AI scoring step can include steps, such as propensity to purchase products or prediction scores for accounts.

Field Name	Description
aiModelConfigStep	**Field Type** string **Description** Required. ID of the AI model config step that's related to the AIScoringStep record.
stepDetail	**Field Type** string **Description** Scoring step details in JSON format.

Declarative Metadata Sample Definition

Here's an example of an AIScoringModelDefinition component.

```
<?xml version="1.0" encoding="UTF-8"?>
<AIScoringModelDefinition xmlns="http://soap.sforce.com/2006/04/metadata">
    <aiModelConfig>Prediction_Scores_for_Accounts</aiModelConfig>
    <aiScoringModelDefVersions>
        <fullName>V1</fullName>
        <aiScoringModelDefinition>Test</aiScoringModelDefinition>
        <aiScoringSteps>

<aiModelConfigStep>Prediction_Scores_for_Accounts.GrainSelector</aiModelConfigStep>
            <stepDetail>{label:Account,name:Account}</stepDetail>
        </aiScoringSteps>
        <aiScoringSteps>

<aiModelConfigStep>Prediction_Scores_for_Accounts.AugmentedDataset</aiModelConfigStep>
        </aiScoringSteps>
        <aiScoringSteps>

<aiModelConfigStep>Prediction_Scores_for_Accounts.TargetConditionBuilder</aiModelConfigStep>

          <stepDetail>{specificOutcomeDefined:Yes,label:Financial accounts are associated
 with an account,name:FA_Target}</stepDetail>
        </aiScoringSteps>
        <aiScoringSteps>

<aiModelConfigStep>Prediction_Scores_for_Accounts.InputVariableSelector</aiModelConfigStep>
```

```
        </aiScoringSteps>
        <aiScoringSteps>

<aiModelConfigStep>Prediction_Scores_for_Accounts.CustomFilter</aiModelConfigStep>
        </aiScoringSteps>
        <aiScoringSteps>

<aiModelConfigStep>Prediction_Scores_for_Accounts.WriteBackConnector</aiModelConfigStep>
        </aiScoringSteps>
        <developerName>V1</developerName>
        <masterLabel>V1</masterLabel>
        <modelMode>TRAIN_AND_DEPLOY</modelMode>
    </aiScoringModelDefVersions>
    <description>Test for metadata</description>
    <masterLabel>Test</masterLabel>
</AIScoringModelDefinition>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>AIScoringModelDefVersion</name>
    </types>
    <types>
        <members>*</members>
        <name>AIScoringModelDefinition</name>
    </types>
    <version>64.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

AIUsecaseDefinition

Represents a collection of fields in your Salesforce org used to define a machine learning use case and get real-time predictions.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

AIUsecaseDefinition components have the suffix `.aiUsecaseDefinitions` and are stored in the `aiUsecaseDefinitions` folder.

Version

AIUsecaseDefinition components are available in API version 56.0 and later.

Special Access Rules

The AIUsecaseDefinition object is available when the admin settings for AI Accelerator and for the product related to the use case are enabled. The Salesforce org must have the CRM Plus license and the product's CRM license.

Fields

Field Name	Description
`aiUsecaseFieldMappings`	**Field Type** AIUsecaseFieldMapping[] **Description** The field mappings for the use case definition. Each use case definition can have multiple field mappings.
`aiUsecaseModels`	**Field Type** AIUsecaseModel[] **Description** The models for the use case definition. Each use case definition can have multiple use case models.
`creatorType`	**Field Type** CreatorType (enumeration of type string) **Description** Required. The type of user who created the use case definition that's used by AI Accelerator. Valid values are: • `INTERNAL_USER` • `SALESFORCE_ADMIN` Available in API version 57.0 and later.
`masterLabel`	**Field Type** string **Description** Required.

Field Name	Description
	A user-friendly name for the use case definition, which is defined when the use case definition is created.
`maximumInsightCount`	**Field Type** int **Description** The maximum number of insights returned by the scoring response.
`maximumRecommendationCount`	**Field Type** int **Description** The maximum number of recommendations returned by the Next Best Action Strategy.
`maximumSuggestionCount`	**Field Type** int **Description** The maximum number of suggestions returned by the scoring response.
`primaryResponseObject`	**Field Type** string **Description** The primary object in which the scoring response is stored based on the specified field mapping.
`recommendationResponseObject`	**Field Type** string **Description** The recommendation response object associated with the use case definition.
`recommendationSource`	**Field Type** RcmdSourceType (enumeration of type string) **Description** The tool or platform that generates recommendations. Valid values are: • `Next_Best_Action_Flow` • `None` Available in API version 57.0 and later.
`secondaryResponseObject`	**Field Type** string **Description** The object in which the scoring response is stored based on the specified field mapping.

Field Name	Description
`shouldSaveFeatures`	**Field Type** boolean **Description** Indicates whether to save the features extracted for the scoring request (`true`) or not (`false`). The default value is `false`.
`shouldSaveInsights`	**Field Type** boolean **Description** Indicates whether to save the prediction insights that are used to generate the score (`true`) or not (`false`). The default value is `false`.
`shouldSaveRecommendation`	**Field Type** boolean **Description** Indicates whether to save the recommendation (`true`) or not (`false`). The default value is `false`.
`shouldSaveRequestResponse`	**Field Type** boolean **Description** Indicates whether to save the request response (`true`) or not (`false`). The default value is `false`.
`shouldSaveScore`	**Field Type** boolean **Description** Indicates whether to save the prediction score (`true`) or not (`false`). The default value is `false`.
`shouldSaveSuggestions`	**Field Type** boolean **Description** Indicates whether to save the suggestions for improving the prediction score (`true`) or not (`false`). The default value is `false`.
`suggestionImpactMinimumPct`	**Field Type** int

Field Name	Description
	Description The minimum eligible percentage for improving the existing prediction score based on the suggestions. Suggestions with an impact greater than the specified percentage on the score are displayed on the prediction scorecard.
`usecaseName`	**Field Type** string **Description** Required. The name of the use case definition.

AIUsecaseFieldMapping

Represents information about the field mapping to store extracted features, prediction scores, prediction insights, and use case suggestions in the response object.

Field Name	Description
`developerName`	**Field Type** string **Description** The unique name for the field mapping in the use case definition. Required. The unique name of the object in the API. This name can contain only underscores and alphanumeric characters, and must be unique in your org. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. In managed packages, this field prevents naming conflicts on package installations. With this field, a developer can change the object's name in a managed package and the changes are reflected in a subscriber's organization. Label is **Record Type Name**.
`mappedFieldName`	**Field Type** string **Description** Required. The name of the field where the scoring response is stored.
`mappedFieldType`	**Field Type** MappedFieldType (enumeration of type string) **Description** Required. The type of the mapped field.

Field Name	Description
	Valid values are: • FEATURE • PREDICTION_SCORE • INSIGHT • SUGGESTION • SECONDARY_RESPONSE_RECORD_ID • RECOMMENDATION_RESPONSE_RECORD_ID • RECOMMENDATION The default value is FEATURE.
masterLabel	**Field Type** string **Description** Required. A user-friendly name for the use case field mapping, which is defined when the field mapping is created.
responseFieldName	**Field Type** string **Description** Required. The name of the response object's field that's mapped to the field storing the score.
responseObject	**Field Type** string **Description** Required. The object whose field is mapped to the field storing the score. It's either the PrimaryResponseObject or the SecondaryResponseObject specified in the AIUsecaseDefinition object.
sequenceNumber	**Field Type** int **Description** The sequence number for the information stored in the field mapping.

AIUsecaseModel

Represents information about the machine learning models that generate predictions for your use case.

Field Name	Description
`aiFeatureExtractors`	**Field Type** AIFeatureExtractor[] **Description** The AI feature extractors to retrieve the input data.
`defaultFeatureExtractor`	**Field Type** AIFeatureExtractor **Description** The default AI feature extractor to retrieve the input data.
`developerName`	**Field Type** string **Description** The unique name for the use case model. Required. The unique name of the object in the API. This name can contain only underscores and alphanumeric characters, and must be unique in your org. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. In managed packages, this field prevents naming conflicts on package installations. With this field, a developer can change the object's name in a managed package and the changes are reflected in a subscriber's organization. Label is **Record Type Name**.
`masterLabel`	**Field Type** string **Description** Required. A user-friendly name for the use case model, which is defined when the use case model is created.
`predictionDefinition`	**Field Type** string **Description** Required. The unique identifier of the prediction definition that's related to the use case model. This identifier can be an external ID. If you use Einstein Discovery to create models, the `predictionDefinition` field stores the developer name of the record.
`predictionPlatform`	**Field Type** PredictionPlatform (enumeration of type string) **Description** Required.

Field Name	Description
	The platform on which the machine learning model is created and deployed. Valid values are: • `Einstein_Discovery` • `Default`—For internal use only. The default value is `Einstein_Discovery`. Available in API version 57.0 and later.

AIFeatureExtractor

Represents information about the feature extractor that's used to retrieve the input data for the use case model that's used to generate predictions.

Field Name	Description
`batchInputSourceIdentifier`	**Field Type** string **Description** The identifier of the input source of the features computed by batch jobs, which can be used by a model for generating predictions. Available in API version 57.0 and later.
`className`	**Field Type** string **Description** Required. The ID of the Apex class created for the feature extractor.
`developerName`	**Field Type** string **Description** The unique name for the feature extractor. Required. The unique name of the object in the API. This name can contain only underscores and alphanumeric characters, and must be unique in your org. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. In managed packages, this field prevents naming conflicts on package installations. With this field, a developer can change the object's name in a managed package and the changes are reflected in a subscriber's organization. Label is **Record Type Name**.
`extractorType`	**Field Type** ExtractorType (enumeration of type string) **Description** Required.

Field Name	Description
	The type of the feature extractor. Valid values are: • `APEX` • `JAVA` • `HYBRID` The default value is `APEX`.
`featureInputType`	**Field Type** string **Description** Required. The type of feature input that's used in generating predictions. Valid values are: • `Realtime_Input` • `Sample_Input` • `Batch_Input` • `Batch_And_Realtime_Input` Available in API version 57.0 and later.
`inputContext`	**Field Type** string **Description** The JSON file with features that act as context for the feature extractor. This data can also include the data in the uploaded CSV file. Available in API version 57.0 and later.
`masterLabel`	**Field Type** string **Description** Required. A user-friendly name for the feature extractor, which is defined when the feature extractor is created.

Declarative Metadata Sample Definition

The following is an example of an AIUsecaseDefinition component.

```
<?xml version="1.0" encoding="UTF-8"?>
<AIUsecaseDefinition xmlns="http://soap.sforce.com/2006/04/metadata">
    <aiUsecaseFieldMappings>
        <developerName>DevName1</developerName>
        <mappedFieldName>Name</mappedFieldName>
```

```
        <mappedFieldType>INSIGHT</mappedFieldType>
        <masterLabel>DevName</masterLabel>
        <responseFieldName>AnnualRevenue</responseFieldName>
        <responseObject>Lead</responseObject>
        <sequenceNumber>2</sequenceNumber>
    </aiUsecaseFieldMappings>
    <aiUsecaseFieldMappings>
        <developerName>DevName2</developerName>
        <mappedFieldName>Value</mappedFieldName>
        <mappedFieldType>INSIGHT</mappedFieldType>
        <masterLabel>DevName</masterLabel>
        <responseFieldName>Id</responseFieldName>
        <responseObject>Account</responseObject>
        <sequenceNumber>2</sequenceNumber>
    </aiUsecaseFieldMappings>
    <aiUsecaseFieldMappings>
        <developerName>DevName3</developerName>
        <mappedFieldName>Score</mappedFieldName>
        <mappedFieldType>PREDICTION_SCORE</mappedFieldType>
        <masterLabel>DevName</masterLabel>
        <responseFieldName>Company</responseFieldName>
        <responseObject>Lead</responseObject>
    </aiUsecaseFieldMappings>
    <aiUsecaseFieldMappings>
        <developerName>DevName4</developerName>
        <mappedFieldName>RecordId</mappedFieldName>
        <mappedFieldType>SECONDARY_RESPONSE_RECORD_ID</mappedFieldType>
        <masterLabel>DevName</masterLabel>
        <responseFieldName>Address</responseFieldName>
        <responseObject>Lead</responseObject>
    </aiUsecaseFieldMappings>
    <aiUsecaseFieldMappings>
        <developerName>DevName1</developerName>
        <mappedFieldName>UsecaseModel1.inputScore</mappedFieldName>
        <mappedFieldType>PREDICTION_SCORE_INPUT</mappedFieldType>
        <masterLabel>DevName</masterLabel>
        <responseFieldName>Score</responseFieldName>
        <responseObject>LeadOutputDMO</responseObject>
        <joinFieldName>LeadId</sequenceNumber>
    </aiUsecaseFieldMappings>
    <aiUsecaseModels>
        <aiFeatureExtractors>
            <className>01pxx0000004X2CAAU</className>
            <extractorType>APEX</extractorType>
            <developerName>DevName2</developerName>
            <masterLabel>DevName</masterLabel>
            <featureInputType>Realtime_Input</featureInputType>
            <inputContext>"{columnNames=[column1, column2], rawData=[S,
315090]}"</inputContext>
            <batchInputSourceIdentifier>DatasetName</batchInputSourceIdentifier>
            <batchInputSourceType>CRMA</batchInputSourceType>
        </aiFeatureExtractors>
        <defaultFeatureExtractor>
            <className>01pxx0000004X0aAAE</className>
```

```
            <extractorType>APEX</extractorType>
            <developerName>DevName1</developerName>
            <masterLabel>DevName</masterLabel>
            <featureInputType>Realtime_Input</featureInputType>
            <inputContext>"{columnNames=[column1, column2], rawData=[S,
315090]}"</inputContext>
            <batchInputSourceIdentifier>DatasetName</batchInputSourceIdentifier>
        </defaultFeatureExtractor>
        <developerName>DevName1</developerName>
        <masterLabel>DevName</masterLabel>
        <predictionDefinition>PredictionDefinitionD</predictionDefinition>
        <predictionPlatform>Einstein_Discovery</predictionPlatform>
        <arePredctGenInRealTime>true</arePredctGenInRealTime>
    </aiUsecaseModels>
    <aiUsecaseModels>
        <developerName>DevName2</developerName>
        <masterLabel>DevName</masterLabel>
        <predictionDefinition>PredictionDefinitionBA</predictionDefinition>
        <predictionPlatform>Einstein_Discovery</predictionPlatform>
        <arePredctGenInRealTime>true</arePredctGenInRealTime>
    </aiUsecaseModels>
    <aiUsecaseModels>
        <developerName>DevName3</developerName>
        <masterLabel>DevName</masterLabel>
        <predictionDefinition>PredictionDefinitionCA</predictionDefinition>
        <predictionPlatform>Einstein_Discovery</predictionPlatform>
        <arePredctGenInRealTime>true</arePredctGenInRealTime>
    </aiUsecaseModels>
    <masterLabel>DevName</masterLabel>
    <maximumInsightCount>3</maximumInsightCount>
    <maximumSuggestionCount>3</maximumSuggestionCount>
    <maximumRecommendationCount>3</maximumRecommendationCount>
    <primaryResponseObject>Lead</primaryResponseObject>
    <secondaryResponseObject>Account</secondaryResponseObject>
    <recommendationResponseObject>Contact</recommendationResponseObject>
    <shouldSaveFeatures>true</shouldSaveFeatures>
    <shouldSaveInsights>true</shouldSaveInsights>
    <shouldSaveRecommendation>false</shouldSaveRecommendation>
    <shouldSaveRequestResponse>false</shouldSaveRequestResponse>
    <shouldSaveScore>true</shouldSaveScore>
    <shouldSaveSuggestions>true</shouldSaveSuggestions>
    <suggestionImpactMinimumPct>50</suggestionImpactMinimumPct>
    <usecaseName>FTestSampleMLUsecase</usecaseName>
    <recommendationSource>Next_Best_Action_Flow</recommendationSource>
    <creatorType>INTERNAL_USER</creatorType>
</AIUsecaseDefinition>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>*AIUsecaseDefinition*</name>
    </types>
```

```
    <version>64.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

AnalyticsDashboard

Represents a Tableau Next dashboard.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

AnalyticsDashboard components have the suffix `.uadash` and are stored in the `analyticsDashboards` folder.

Version

AnalyticsDashboard components are available in API version 64.0 and later.

Fields

Field Name	Description
`analyticsWorkspace`	**Field Type** string **Description** Required. The Tableau Next workspace the dashboard belongs to.
`description`	**Field Type** string **Description** The description of the dashboard.
`layouts`	**Field Type** AnalyticsDashboardLayout[]

Field Name	Description
	Description The layouts for the dashboard. A dashboard has 0 or more layouts.
`masterLabel`	**Field Type** string **Description** Required. The name of the dashboard.
`style`	**Field Type** string **Description** The style of the dashboard. This is a JSON string. Example: `{"widgetStyle":{"backgroundColor":"#ffffff","borderRadius":[],"borderColor":"#cccccc","borderWidth":1,"borderRadius":0}}`
`templateAssetSourceName`	**Field Type** string **Description** If the dashboard was created from a template, the name of the source dashboard in the template.
`templateSource`	**Field Type** string **Description** If the dashboard was created from a template, the name of the source template.
`version`	**Field Type** double **Description** The API version of the dashboard.
`widgets`	**Field Type** AnalyticsDashboardWidget[] **Description** A list of widgets in the dashboard. A dashboard has 0 or more widgets.
`workspaceAssetRelationships`	**Field Type** AnalyticsWorkspaceAsset[]

Field Name	Description
	Description A list of analytics assets in the workspace this dashboard is associated with. A dashboard has 0 or more workspace asset relationships.

AnalyticsDashboardLayout

Represents a layout for a Tableau Next dashboard. A dashboard can have multiple layouts, like Desktop and Mobile.

Field Name	Description
`analyticsDashboard`	**Field Type** string **Description** Required. The name of the dashboard the layout is associated with.
`columnCount`	**Field Type** string **Description** The number of columns in the layout.
`label`	**Field Type** string **Description** The label for the layout.
`layoutName`	**Field Type** string **Description** The name of the layout.
`maxWidth`	**Field Type** string **Description** The max width of the layout, in pixels.
`pages`	**Field Type** AnalyticsDashboardPage[] **Description** The pages to display for this dashboard layout. A layout has 0 or more pages.

Field Name	Description
rowHeight	**Field Type** string **Description** The row height for layout rows.
style	**Field Type** string **Description** The style for the layout. This is a JSON string. Example: `{"backgroundColor":"#ffffff","gutterColor":"#f3f3f3","cellSpacingX":8,"cellSpacingY":8`
version	**Field Type** double **Description** The API version of the dashboard layout.

AnalyticsDashboardPage

Represents a page in a Tableau Next dashboard.

Field Name	Description
index	**Field Type** int **Description** The index of the page in the dashboard. An index of `0` is the first page of the dashboard. No index is required if there is only the default page of the dashboard.
label	**Field Type** string **Description** The label for the dashboard page.
pageName	**Field Type** string **Description** The generated unique ID for the dashboard page.
pageWidgets	**Field Type** AnalyticsDashPageWidget[]

Field Name	Description
	Description A list of dashboard page widgets. A page has 0 or more page widgets.

AnalyticsDashPageWidget

Represents an dashboard page widget for a Tableau Next dashboard page.

Field Name	Description
`analyticsDashboardWidget`	**Field Type** string **Description** Required. The name of dashboard widget, defined in the list of widgets for the dashboard.
`colspan`	**Field Type** string **Description** The column span for the widget on the page. This is a numeric value.
`column`	**Field Type** string **Description** The column the widget is placed in on the page. This is a numeric value.
`row`	**Field Type** string **Description** The row the widget is placed in on the page. This is a numeric value.
`rowspan`	**Field Type** string **Description** The row span for the widget on the page. This is a numeric value.

AnalyticsDashboardWidget

Represents a widget on a Tableau Next dashboard.

Field Name	Description
`analyticsDashboard`	**Field Type** string **Description** Required. The API name of the dashboard the widget is associated with.
`buttonWidgetDefs`	**Field Type** AnalyticsButtonWidgetDef[] **Description** The definition for a button widget type. A dashboard has 0 or more button widgets. Required for button widget.
`containerWidgetDefs`	**Field Type** AnalyticsContainerWidgetDef[] **Description** The definition for a container widget type. A dashboard has 0 or more container widgets. Required for container widget.
`filterWidgetDefs`	**Field Type** AnalyticsFilterWidgetDef[] **Description** The definition for a filter widget type. A dashboard has 0 or more filter widgets. Required for filter widget.
`label`	**Field Type** string **Description** The label for the widget.
`metricWidgetDefs`	**Field Type** AnalyticsMetricWidgetDef[] **Description** The definition for a metric widget type. A dashboard has 0 or more metric widgets. Required for metric widget.
`parameterWidgetDefs`	**Field Type** AnalyticsParamWidgetDef[] **Description** The definition for a parameter widget type. A dashboard has 0 or more paramet widgets. Required for parameter widget.

Field Name	Description
`textWidgetDefs`	**Field Type** AnalyticsTextWidgetDef[] **Description** The definition for a text widget type. A dashboard has 0 or more text widgets. Required for text widget.
`type`	**Field Type** AnalyticsWidgetType (enumeration of type string) **Description** Required. The widget type. Values are: • `button` • `container` • `filter` • `metric` • `parameter` • `text` • `visualization`
`vizWidgetDefs`	**Field Type** AnalyticsVizWidgetDef[] **Description** The definition for a visualization widget type. A dashboard has 0 or more visualization widgets. Required for visualization widget.
`widgetActions`	**Field Type** AnalyticsAssetAction[] **Description** The actions for the widget. A widget has 0 or more text widgets.
`widgetName`	**Field Type** string **Description** The API name of the widget. Use this for the `analyticsDashboardWidget` value in `AnalyticsDashPageWidget`.

AnalyticsAssetAction

Represents an action for a Tableau Next asset.

Field Name	Description
`actionType`	**Field Type** AnalyticsActionType (enumeration of type string) **Description** Required. The action type. Values are: • `flow` • `navigate` • `recordaction`
`eventType`	**Field Type** AnalyticsActionEventType (enumeration of type string) **Description** Required. The action event type. Values are: • `click` • `select`
`parameters`	**Field Type** string **Description** The parameter for the action. This is a JSON string. Example: `{"destination":{"type":"url","target":"www.salesforce.com"}}`
`version`	**Field Type** double **Description** The API version of the action.

AnalyticsButtonWidgetDef

Represents a button widget definition for a Tableau Next dashboard.

Field Name	Description
`parameters`	**Field Type** string

Field Name	Description
	Description The parameters for the button widget. This is a JSON String. Example: `{"text":"Button","alignmentX":"center","alignmentY":"center","fontSize":16}`

AnalyticsContainerWidgetDef

Represents a container widget definition for a Tableau Next dashboard.

Field Name	Description
`parameters`	**Field Type** string **Description** The parameters for the container widget. This is a JSON String. Example: `{"widgetStyle":{"backgroundColor":"#1295FF","borderEdges":[]}}`

AnalyticsFilterWidgetDef

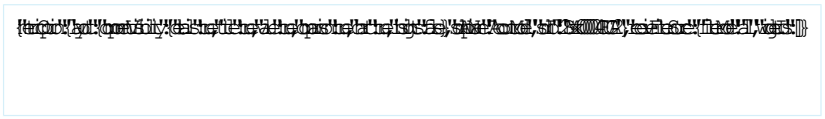
Represents a filter widget definition for a Tableau Next dashboard.

Field Name	Description
`initialValues`	**Field Type** string **Description** The initial values for the filter.
`parameters`	**Field Type** string **Description** The parameters for the filter widget. This is a JSON String. Example: `{"widgetStyle":{"borderEdges":[]},"filterDefinition":{"objectApiName":"Account","fieldApiName":"Account_Type__c","dataType":"text","selectionType":"multi","isRequired":false}}`

Field Name	Description
source	**Field Type** string **Description** The data source to apply the filter to.

AnalyticsMetricWidgetDef

Represents a metric widget definition for a Tableau Next dashboard.

Field Name	Description
parameters	**Field Type** string **Description** The parameters for the filter widget. This is a JSON String. Example: 
source	**Field Type** string **Description** The data source the metric is associated with.
version	**Field Type** double **Description** The API version of the metric.

AnalyticsParamWidgetDef

Represents a parameters widget definition for a Tableau Next dashboard.

Field Name	Description
initialValues	**Field Type** string **Description** The initial values for the parameters.

Field Name	Description
parameters	**Field Type** string **Description** The parameters for the filter widget. This is a JSON String. Example: `{"parameterName":"AccountParameter_prm","isLabelHidden":false}`
source	**Field Type** string **Description** The data source the parameters are associated with.

AnalyticsTextWidgetDef

Represents a text widget definition for a Tableau Next dashboard.

Field Name	Description
parameters	**Field Type** string **Description** The parameters for the filter widget. This is a JSON String. Example: `{"content":[{"attributes":{"color":"#000000","size":"12px"},"insert":"full dashboard"},{"attributes":{"align":"left"},"insert":"\n"}]}`

AnalyticsVizWidgetDef

Represents a visualization widget definition for a Tableau Next dashboard.

Field Name	Description
analyticsVisualization	**Field Type** string **Description** The API name of the visualization.

Field Name	Description
parameters	**Field Type** string **Description** The parameters for the filter widget. This is a JSON String. Example: `{"legendPosition":"Right","receiveFilterSource":{"filterMode":"all","widgetIds":[]}}`

Declarative Metadata Sample Definition

The following is an example of an AnalyticsDashboard component.

```
<?xml version="1.0" encoding="UTF-8"?>
<AnalyticsDashboard xmlns="http://soap.sforce.com/2006/04/metadata"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <analyticsWorkspace>My_Workspace</analyticsWorkspace>
    <description>A dashboard</description>
    <layouts>
        <analyticsDashboard>My_Dashboard</analyticsDashboard>
        <columnCount>36</columnCount>
        <layoutName>default</layoutName>
        <maxWidth>1200</maxWidth>
        <label>layoutLabel</label>
        <pages>
            <index>0</index>
            <label>Page 1</label>
            <pageName>e412bc11-d43b-4fba-ab44-b31bb842b49a</pageName>
            <pageWidgets>
                <analyticsDashboardWidget>visualization_1</analyticsDashboardWidget>
                <colspan>11</colspan>
                <column>1</column>
                <row>2</row>
                <rowspan>10</rowspan>
            </pageWidgets>
            <pageWidgets>
                <analyticsDashboardWidget>button_1</analyticsDashboardWidget>
                <colspan>9</colspan>
                <column>13</column>
                <row>8</row>
                <rowspan>2</rowspan>
            </pageWidgets>
            <pageWidgets>
                <analyticsDashboardWidget>container_1</analyticsDashboardWidget>
                <colspan>11</colspan>
                <column>23</column>
                <row>2</row>
                <rowspan>10</rowspan>
            </pageWidgets>
            <pageWidgets>
```

```
                <analyticsDashboardWidget>text_2</analyticsDashboardWidget>
                <colspan>9</colspan>
                <column>13</column>
                <row>5</row>
                <rowspan>2</rowspan>
            </pageWidgets>
            <pageWidgets>
                <analyticsDashboardWidget>metric_1</analyticsDashboardWidget>
                <colspan>11</colspan>
                <column>1</column>
                <row>13</row>
                <rowspan>10</rowspan>
            </pageWidgets>
            <pageWidgets>
                <analyticsDashboardWidget>list_4</analyticsDashboardWidget>
                <colspan>9</colspan>
                <column>13</column>
                <row>13</row>
                <rowspan>2</rowspan>
            </pageWidgets>
            <pageWidgets>
                <analyticsDashboardWidget>list_1</analyticsDashboardWidget>
                <colspan>9</colspan>
                <column>13</column>
                <row>2</row>
                <rowspan>2</rowspan>
            </pageWidgets>
        </pages>
        <rowHeight>24</rowHeight>

<style>{&qut;backgroundColor&qut;:&qut;#ffffff&qut;,&qut;gutterColor&qut;:&qut;#f3f3f3&qut;,&qut;cellSpacingX&qut;:8,&qut;cellSpacingY&qut;:8}</style>

    </layouts>
    <masterLabel>My Dashboard</masterLabel>

<style>{&qut;widgetStyle&qut;:{&qut;backgroundColor&qut;:&qut;#ffffff&qut;,&qut;borderEdges&qut;:[],&qut;borderColor&qut;:&qut;#cccccc&qut;,&qut;borderWidth&qut;:1,&qut;borderRadius&qut;:0}}</style>

    <version>64.0</version>
    <widgets>
        <analyticsDashboard>My_Dashboard</analyticsDashboard>
        <type>visualization</type>
        <vizWidgetDefs>
            <analyticsVisualization>New_Visualization</analyticsVisualization>

<parameters>{&qut;legendPosition&qut;:&qut;Right&qut;,&qut;receiveFilterSource&qut;:{&qut;filterMode&qut;:&qut;all&qut;,&qut;widgetIds&qut;:[]}}</parameters>

        </vizWidgetDefs>
        <widgetName>visualization_1</widgetName>
    </widgets>
    <widgets>
        <analyticsDashboard>My_Dashboard</analyticsDashboard>
        <buttonWidgetDefs>

<parameters>{&qut;text&qut;:&qut;Button&qut;,&qut;alignmentX&qut;:&qut;center&qut;,&qut;alignmentY&qut;:&qut;center&qut;,&qut;fontSize&qut;:16}</parameters>
```

```
        </buttonWidgetDefs>
        <type>button</type>
        <widgetActions>
            <actionType>navigate</actionType>
            <eventType>click</eventType>

<parameters>{&quot;destination&quot;:{&quot;type&quot;:&quot;url&quot;,&quot;target&quot;:&quot;www.salesforce.com&quot;}}</parameters>

            <version>63.0</version>
        </widgetActions>
        <widgetName>button_1</widgetName>
    </widgets>
    <widgets>
        <analyticsDashboard>My_Dashboard</analyticsDashboard>
        <containerWidgetDefs>

<parameters>{&quot;widgetStyle&quot;:{&quot;backgroundColor&quot;:&quot;#1295FF&quot;,&quot;borderEdges&quot;:[]}}</parameters>

        </containerWidgetDefs>
        <type>container</type>
        <widgetName>container_1</widgetName>
    </widgets>
    <widgets>
        <analyticsDashboard>My_Dashboard</analyticsDashboard>
        <textWidgetDefs>

<parameters>{&quot;content&quot;:[{&quot;attributes&quot;:{&quot;color&quot;:&quot;#000000&quot;,&quot;size&quot;:&quot;12px&quot;},&quot;insert&quot;:&quot;full

dashboard&quot;},{&quot;attributes&quot;:{&quot;align&quot;:&quot;left&quot;},&quot;insert&quot;:&quot;\n&quot;}]}</parameters>

        </textWidgetDefs>
        <type>text</type>
        <widgetName>text_2</widgetName>
    </widgets>
    <widgets>
        <analyticsDashboard>My_Dashboard</analyticsDashboard>
        <metricWidgetDefs>

<parameters>{&quot;aggregates&quot;:[{&quot;calculatedFormula&quot;:&quot;RowCount&quot;,&quot;datapoints&quot;:[{&quot;compareValues&quot;:[],&quot;filters&quot;:[{&quot;column&quot;:&quot;Account.Type&quot;,&quot;operator&quot;:&quot;equals&quot;,&quot;values&quot;:[&quot;Customer - Direct&quot;]}],&quot;label&quot;:&quot;Customer - Direct&quot;},{&quot;compareValues&quot;:[],&quot;filters&quot;:[{&quot;column&quot;:&quot;Account.Type&quot;,&quot;operator&quot;:&quot;equals&quot;,&quot;values&quot;:[&quot;Customer - Channel&quot;]}],&quot;label&quot;:&quot;Customer - Channel&quot;},{&quot;compareValues&quot;:[],&quot;filters&quot;:[{&quot;column&quot;:&quot;Account.Type&quot;,&quot;operator&quot;:&quot;equals&quot;,&quot;values&quot;:[&quot;Channel Partner / Reseller&quot;]}],&quot;label&quot;:&quot;Channel Partner / Reseller&quot;},{&quot;compareValues&quot;:[],&quot;filters&quot;:[{&quot;column&quot;:&quot;Account.Type&quot;,&quot;operator&quot;:&quot;equals&quot;,&quot;values&quot;:[&quot;Installation Partner&quot;]}],&quot;label&quot;:&quot;Installation Partner&quot;},{&quot;compareValues&quot;:[],&quot;filters&quot;:[{&quot;column&quot;:&quot;Account.Type&quot;,&quot;operator&quot;:&quot;equals&quot;,&quot;values&quot;:[&quot;Technology Partner&quot;]}],&quot;label&quot;:&quot;Technology Partner&quot;},{&quot;compareValues&quot;:[],&quot;filters&quot;:[{&quot;column&quot;:&quot;Account.Type&quot;,&quot;operator&quot;:&quot;equals&quot;,&quot;values&quot;:[&quot;Other&quot;]}],&quot;label&quot;:&quot;Other&quot;}],&quot;label&quot;:&quot;Record Count&quot;,&quot;name&quot;:&quot;RowCount&quot;}],&quot;groupings&quot;:[],&quot;standardDateFilter&quot;:{&quot;column&quot;:&quot;CUSTOM_DATE&quot;,&quot;durationValue&quot;:&quot;CURRENT_FISCAL_QUARTER&quot;,&quot;startDate&quot;:null,&quot;endDate&quot;:null},&quot;timeFrameFilter&quot;:{&quot;dateColumn&quot;:&quot;Account.CreatedDate&quot;,&quot;interval&quot;:&quot;INTERVAL_CURRENT&quot;,&quot;period&quot;:&quot;Quarter&quot;}}</parameters>

            <source>AccountMetric_mtc</source>
        </metricWidgetDefs>
        <type>metric</type>
        <widgetName>metric_1</widgetName>
    </widgets>
    <widgets>
        <analyticsDashboard>My_Dashboard</analyticsDashboard>
        <filterWidgetDefs>
            <initialValues>null</initialValues>

<parameters>{&quot;aggregates&quot;:[],&quot;groupings&quot;:[{&quot;dateGranularity&quot;:&quot;None&quot;,&quot;name&quot;:&quot;Account.Type&quot;,&quot;sortAggregate&quot;:null,&quot;sortOrder&quot;:null}],&quot;standardDateFilter&quot;:{&quot;column&quot;:&quot;CUSTOM_DATE&quot;,&quot;durationValue&quot;:&quot;CURRENT_FISCAL_QUARTER&quot;,&quot;startDate&quot;:null,&quot;endDate&quot;:null},&quot;timeFrameFilter&quot;:{&quot;dateColumn&quot;:&quot;Account.CreatedDate&quot;,&quot;interval&quot;:&quot;INTERVAL_CURRENT&quot;,&quot;period&quot;:&quot;Quarter&quot;}}</parameters>

            <source>AccountModel</source>
```

```
        </filterWidgetDefs>
        <label>Account Id</label>
        <type>filter</type>
        <widgetName>list_4</widgetName>
    </widgets>
    <widgets>
        <analyticsDashboard>My_Dashboard</analyticsDashboard>
        <label>AccountParameter</label>
        <parameterWidgetDefs>
            <initialValues>null</initialValues>

<parameters>{&quot;parameterName&quot;:&quot;AccountParameter_prm&quot;,&quot;isLabelHidden&quot;:false}</parameters>

            <source>AccountModel</source>
        </parameterWidgetDefs>
        <type>parameter</type>
        <widgetName>list_1</widgetName>
    </widgets>
    <templateSource></templateSource>
    <templateAssetSourceName></templateAssetSourceName>
    <workspaceAssetRelationships>
        <asset xsi:nil="true"/>
        <assetType>AnalyticsDashboard</assetType>
        <assetUsageType>Created</assetUsageType>
        <workspace>My_Workspace</workspace>
    </workspaceAssetRelationships>
</AnalyticsDashboard>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>My_Dashboard</members>
        <name>AnalyticsDashboard</name>
    </types>
    <version>64.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

AnalyticSnapshot

Represents a reporting snapshot. A reporting snapshot lets you report on historical data. Authorized users can save tabular or summary report results to fields on a custom object, then map those fields to corresponding fields on a target object. They can then schedule when to run the report to load the custom object's fields with the report's data. Reporting snapshots enable you to work with report data similarly to how you work with other records in Salesforce.

Declarative Metadata File Suffix and Directory Location

Lightning Platform AnalyticSnapshot components are stored in the `analyticSnapshots` directory of the corresponding package directory. The file name matches the unique name of the reporting snapshot, and the extension is `.snapshot`.

Version

Lightning Platform AnalyticSnapshot components are available in API version 16.0 and later.

Fields

Field	Field Type	Description
`description`	string	A description of the reporting snapshot.
`fullName`	string	The reporting snapshot name used for API access. The name can only contain characters, letters, and the underscore (_) character. It must start with a letter and can't end with an underscore or contain two consecutive underscore characters. This field is inherited from the Metadata component.
`groupColumn`	string	A column that specifies which level to extract data from the source report. It's only applicable for summary reports.
`mappings`	AnalyticSnapshotMapping[]	A list of reporting snapshot mappings. For valid values, see AnalyticSnapshotMapping.
`name`	string	Required. The display name of the reporting snapshot.
`runningUser`	string	The username of the user whose role and *sharing* settings are used to run the reporting snapshot.
`sourceReport`	string	Required. The report where data is extracted from.
`targetObject`	string	Required. The custom object where data is inserted.

AnalyticSnapshotMapping

AnalyticSnapshotMapping defines the mapping for the reporting snapshot. Valid values are:

Field	Field Type	Description
`aggregateType`	ReportSummaryType[] (enumeration of type string)	List that defines if and how each report field is summarized. For valid values, see ReportSummaryType.
`sourceField`	string	The sourceField can be one of the following: • The field on the sourceReport that you want to map to the targetField in the targetObject • A summary of a filed on the sourceReport (for Summary reports only)

Field	Field Type	Description
		• A field on the reporting snapshot, such as JobName, RunningUser, or ExecutionTime (set through the user interface) **Note:** The sourceField must correspond to the sourceType you specify.
`sourceType`	ReportJobSourceTypes[] (enumeration of type string)	List that defines the report format for the reporting snapshot. For valid values, see ReportJobSourceTypes.
`targetField`	string	A field on the targetObject into which this particular sourceField is inserted.

ReportJobSourceTypes

An enumeration of type string that defines the report format for the reporting snapshot. Valid values are:

Enumeration Value	Description
`snapshot`	Use this option if the sourceField contains snapshot-specific information such as JobName, RunningUser, or ExecutionTime.
`summary`	Use this option if referencing a summary (Sum, Average, Minimum, Maximum) of a field from the sourceReport.
`tabular`	Use this option if referencing an available column from the sourceReport.

Declarative Metadata Sample Definition

Here's a sample XML definition of a reporting snapshot.

```
<?xml version="1.0" encoding="UTF-8"?>
<AnalyticSnapshot xmlns="http://soap.sforce.com/2006/04/metadata">
    <description>my description</description>
    <groupColumn>INDUSTRY</groupColumn>
    <mappings>
        <aggregateType>Average</aggregateType>
        <sourceField>SALES</sourceField>
        <sourceType>summary</sourceType>
        <targetField> myObject __c.Name</targetField>
    </mappings>
    <mappings>
        <sourceField>ExecutionTime</sourceField>
        <sourceType>snapshot</sourceType>
        <targetField> myObject __c.field3__c</targetField>
    </mappings>
    <mappings>
        <sourceField>INDUSTRY</sourceField>
        <sourceType>tabular</sourceType>
        <targetField>testObject__c.Name</targetField>
    </mappings>
    <name>my snapshot</name >
    <runningUser>user@salesforce.com</runningUser>
```

```
    <sourceReport>myFolder/mytSummaryReport</sourceReport>
    <targetObject>myObject__c</targetObject>
</AnalyticSnapshot>
```

Wildcard Support in the Manifest File

This metadata type doesn't support the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SEE ALSO:

Report

AnalyticsVisualization

Represents a Tableau Next visualization.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

AnalyticsVisualization components have the suffix `.uaviz` and are stored in the `analyticsVisualizations` folder.

Version

AnalyticsVisualization components are available in API version 64.0 and later.

Fields

Field Name	Description
`actions`	**Field Type** AnalyticsAssetAction[] **Description** The actions for the visualization. A visualization has 0 or more actions.
`analyticsWorkspace`	**Field Type** string **Description** Required. The Tableau Next workspace the visualization belongs to.

Field Name	Description
`dataSource`	**Field Type** string **Description** Required. The data source for the visualization.
`fields`	**Field Type** AnalyticsVizField[] **Description** A list of data fields for the visualization. A visualization has 0 or more fields.
`masterLabel`	**Field Type** string **Description** Required. The name of the visualization.
`templateAssetSourceName`	**Field Type** string **Description** If the visualization was created from a template, the name of the source visualization in the template.
`templateSource`	**Field Type** string **Description** If the visualization was created from a template, the name of the source template.
`version`	**Field Type** double **Description** The API version of the visualization.
`views`	**Field Type** AnalyticsVizViewDef[] **Description** A list of views for the visualization. A visualization has 0 or more views.
`visualSpecification`	**Field Type** base64Binary

Field Name	Description
	Description Required. The visual specification for the visualization.
`workspaceAssetRelationships`	**Field Type** AnalyticsWorkspaceAsset[] **Description** A list of analytics assets in the workspace this visualization is associated with. A visualization has 0 or more workspace asset relationships.

AnalyticsVizField

Represents a data field in a visualization.

Field Name	Description
`computeUsing`	**Field Type** string **Description** The expression to compute the field value with.
`displayCategory`	**Field Type** VisualizationFieldDisplayCategoryType (enumeration of type string) **Description** The display category type for the visualization field. Values are: • `Continuous` • `Discrete`
`fieldKey`	**Field Type** string **Description** Required. The key for the field.
`fieldName`	**Field Type** string **Description** The name of the field.

Field Name	Description
function	**Field Type** VisualizationFieldFunctionType (enumeration of type string) **Description** The function type of the visualization field. Values are: • Avg • Count • CountD • DatePartDay • DatePartMonth • DatePartQuarter • DatePartWeek • DatePartWeekDay • DatePartYear • DateTruncDay • DateTruncMonth • DateTruncQuarter • DateTruncWeek • DateTruncYear • FiscalDatePartMonth • FiscalDatePartQuarter • FiscalDatePartWeek • FiscalDatePartYear • FiscalDateTruncMonth • FiscalDateTruncQuarter • FiscalDateTruncWeek • FiscalDateTruncYear • Max • Mdy • Median • Min • My • Stdev • Stdevp • Sum • UserAgg • Var • Varp

Field Name	Description
`label`	**Field Type** string **Description** The label for the field.
`objectName`	**Field Type** string **Description** The name of the data source object for the field.
`quickTableCalc`	**Field Type** string **Description** The expression to do a quick table calculation with.
`role`	**Field Type** VisualizationFieldRoleType (enumeration of type string) **Description** The role type of the visualization field. Values are: • `Dimension` • `Measure`
`type`	**Field Type** VisualizationFieldType (enumeration of type string) **Description** The type of the visualization field. Values are: • `Field` • `MeasureNames` • `MeasureValues`

AnalyticsVizViewDef

Represents a view definition for a Tableau Next visualization.

Field Name	Description
`fullName`	**Field Type** string

Field Name	Description
	Description Required. The full name of the view definition.
`isOriginal`	**Field Type** boolean **Description** Required. Indicates whether the view is original (`true`) or not (`false`).
`masterLabel`	**Field Type** string **Description** Required. The name of the view definition.
`version`	**Field Type** double **Description** The API version of the visualization view.
`viewSpecification`	**Field Type** string **Description** The specification for the view definition.

Declarative Metadata Sample Definition

The following is an example of an AnalyticsVisualization component.

```
<?xml version="1.0" encoding="UTF-8"?>
<AnalyticsVisualization xmlns="http://soap.sforce.com/2006/04/metadata"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <analyticsWorkspace>My_Workspace</analyticsWorkspace>
  <description>A visualization</description>
  <dataSource>My_Semantic_Model</dataSource>
  <fields>
    <displayCategory>Discrete</displayCategory>
    <fieldKey>Field1</fieldKey>
    <fieldName>Lead_Source18</fieldName>
    <objectName>Opportunity_Home</objectName>
    <role>Dimension</role>
     <type>Field</type>
```

```
  </fields>
  <fields>
    <fieldKey>Field2</fieldKey>
    <displayCategory>Continuous</displayCategory>
    <fieldName>Amount</fieldName>
    <function>Sum</function>
    <objectName>Opportunity_Home</objectName>
    <role>Measure</role>
    <type>Field</type>
  </fields>
  <views>
    <fullName>default</fullName>
    <masterLabel>My_Visualization_default</masterLabel>
    <viewSpecification>

    <isOriginal>true</isOriginal>
  </views>
  <masterLabel>My_Visualization</masterLabel>
  <version>64.0</version>
  <templateSource></templateSource>
  <templateAssetSourceName></templateAssetSourceName>
  <workspaceAssetRelationships>
    <asset xsi:nil="true"/>
    <assetType>AnalyticVisualization</assetType>
    <assetUsageType>Created</assetUsageType>
    <workspace>My_Workspace</workspace>
  </workspaceAssetRelationships>
</AnalyticsVisualization>
```

The following is an example `package.xml` that references the metadata definition.

```
<?xml version="1.0" encoding="UTF-8"?>
 <Package xmlns="http://soap.sforce.com/2006/04/metadata">
 <types>
 <members>My_Visualization</members>
 <name>AnalyticsVisualization</name>
 </types>
 <version>64.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

AnalyticsWorkspace

Represents a Tableau Next workspace.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

AnalyticsWorkspace components have the suffix `.analtyicsWorkspace` and are stored in the `analyticsWorkspaces` folder.

Version

AnalyticsWorkspace components are available in API version 64 and later.

Fields

Field Name	Description
`description`	**Field Type** string **Description** The workspace description.
`masterLabel`	**Field Type** string **Description** Required. The name of the workspace.
`workspaceAssetRelationships`	**Field Type** AnalyticsWorkspaceAsset[] **Description** The workspace assets associated with the workspace. A workspace has 1 or more assets.

AnalyticsWorkspaceAsset

Represents a Tableau Next analytics asset.

Field Name	Description
`asset`	**Field Type** string

Field Name	Description
	Description Required. The name of workspace asset
`assetType`	**Field Type** AnalyticsWorkspaceAssetType (enumeration of type string) **Description** Required. The workspace asset type Values are: • `AnalyticsDashboard` (Tableau Next Dashboard) • `AnalyticsVisualization` (Tableau Next Visualization) • `MktCalculatedInsightObject` (Data Cloud Calculated Insight Object) • `MktDataLakeObject` (Data Cloud Data Lake Object) • `MktDataModelObject` (Data Cloud Data Model Object) • `SemanticModel` (Semantic Model)
`assetUsageType`	**Field Type** AnalyticsWorkspaceAssetUsageType (enumeration of type string) **Description** Required. The workspace asset usage type. Values are: • `Created` • `Referenced`
`workspace`	**Field Type** string **Description** Required. The workspace the asset belongs to.

Declarative Metadata Sample Definition

The following is an example of an AnalyticsWorkspace component.

```
<?xml version="1.0" encoding="UTF-8"?>
  <AnalyticsWorkspace xmlns="http://soap.sforce.com/2006/04/metadata">
```

```
  <description>An example for Analytics Workspace</description>
  <masterLabel>Analytics Workspace</masterLabel>
  <workspaceAssetRelationships>
     <asset>My Test Dashboard</asset>
     <assetType>AnalyticsDashboard</assetType>
     <assetUsageType>Created</assetUsageType>
     <workspace>Analytics Workspace</workspace>
  </workspaceAssetRelationhips>
</AnalyticsWorkspace>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
  <types>
    <members>*</members>
    <name>AnalyticsWorkspace</name>
  </types>
  <version>64.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

AnimationRule

Represents criteria for determining when an animation is displayed to Path users.This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

AnimationRule components have the suffix `animationRule` and are stored in the `animationRules` folder.

Version

AnimationRule components are available in API version 46.0 and later.

Fields

Field Name	Field Type	Description
`animationFrequency`	picklist	Required. The frequency with which an animation is displayed when a user selects the designated picklist values in a path. Valid values are: • `always` • `often` • `sometimes` • `rarely` A value of `always` triggers an animation every time. The values `often`, `sometimes`, and `rarely` trigger an animation progressively less frequently.
`developerName`	string	Required. The developer name for the animation rule. Note: Only users with View DeveloperName OR View Setup and Configuration permission can view, group, sort, and filter this field.
`isActive`	boolean	Required. Indicates whether the animation rule is active (`true`) or not (`false`).
`masterLabel`	string	Required. The label for the animation rule.
`recordTypeContext`	picklist	Required. An enum to track whether this AnimationRule applies to all record types for the associated sObject, or only to a single or main record type. Valid values are `All`, `Master`, or `Custom`.
`recordTypeName`	reference	The record type selected for the sObject in which the animation is displayed.
`sobjectType`	string	The object on which the animation rule is run.
`targetField`	string	Required. Name of the field used to determine when to display an animation.
`targetFieldChangeToValues`	string	Required. Values used to determine when to display an animation. When a user selects a value in `targetField` that matches a value stored in `targetFieldChangeToValues`, the animation is displayed.

Declarative Metadata Sample Definition

The following is an example of an AnimationRule component.

```
<?xml version="1.0" encoding="UTF-8"?>
<AnimationRule xmlns="http://soap.sforce.com/2006/04/metadata">
    <animationFrequency>Always</animationFrequency>
    <developerName>AnimationRule_DeveloperName</developerName>
    <isActive>true</isActive>
    <masterLabel>AnimationRule Label</masterLabel>
```

```
    <recordTypeContext>All</recordTypeContext>
    <recordTypeName>__MASTER__</recordTypeName>
    <sobjectType>Opportunity</sobjectType>
    <targetField>StageName</targetField>
   <targetFieldChangeToValues>Delivered, Negotiating, Closed Won</targetFieldChangeToValues>
</AnimationRule>
```

The following is an example `package.xml` that references the AnimationRule component.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>PathAssistant</members>
        <name>Settings</name>
    </types>
    <types>
        <members>AnimationRule_Developer_Name</members>
        <name>AnimationRule</name>
    </types>
    <version>46.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

AppFrameworkTemplateBundle

Represents the app framework template bundle. Use these templates for Data Cloud and Tableau Next assets.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

An app framework template bundle is a folder that contains definition files for a template. Unlike other metadata components, a AppFrameworkTemplateBundle component isn't represented by a single component file, but instead by a collection of JSON and other definition files. Each definition file represents a resource in a template, such as semantic models, workspaces, visualizations, and dashboasrds. For example, this directory structure shows the hierarchy of the folders and files for one App Framework Template definition, myTemplate.

```
appTemplates
  myTemplate
    template-info.json
    create-chain.json
    rules.json
    variables.json
    layout.json
```

```
    workspaces
      myWorkspace.json
    dashboards
      myDashboard.json
```

App framework template bundles must be under a top-level folder that's named `appTemplates`. Each bundle must have its own subfolder under the `appTemplate` folder and be named withthe template's fully qualified API name. The bundle folder must contain a template-info.json file to specify the metadata about the template and the references to other definition files. An entire bundle doesn't have a suffix and definition files can have one of the following suffixes.

Suffix	Component Type
.json	JSON file
.html	HTML file

Version

AppFrameworkTemplateBundle components are available in API version 64.0 and later.

Special Access Rules

Definitions can be created in both managed and unmanaged packages.

Fields

Field Name	Description
`assetVersion`	**Field Type** double **Description** The API version of the app template bundle.
`description`	**Field Type** string **Description** The description for the app template.
`label`	**Field Type** string **Description** Required The label for app template.
`maxAppCount`	**Field Type** int

Field Name	Description
	Description The maximum number of apps that can be created from this app template.
`templateBadgeIcon`	**Field Type** string **Description** The badge icon for app template. This must be a `.png` file type.
`templateType`	**Field Type** string **Description** The type for app template.

Declarative Metadata Sample Definition

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
<types>
    <members>myTemplate</members>
    <name>AppFrameworkTemplateBundle</name>
</types>
<version>64.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ArticleType

Represents the metadata associated with an article type.

All articles in Salesforce Knowledge are assigned to an *article type*. An article's type determines the type of content it contains, its appearance, and which users can access it. For example, a simple FAQ article type can have two custom fields, `Question` and `Answer`, where article managers enter data when creating or updating FAQ articles. A more complex article type can have dozens of fields organized into several sections. Using layouts and templates, administrators can structure the article type in the most effective way for its particular content. User access to article types is controlled by permissions. For each article type, an administrator can grant "Create," "Read," "Edit," or "Delete" permissions to users. For example, the article manager can allow internal users to read, create, and edit FAQ article types, but let partner users only read FAQs. See "Knowledge Article Types" in the Salesforce online help and Knowledge in the *SOAP API Developer Guide*.

Declarative Metadata File Suffix and Directory Location

An ArticleType is defined as a custom object and is stored in the `objects` folder. ArticleTypes have a suffix `__kav` (instead of `__c` for custom objects). ArticleType field names have a suffix of `__c` like other custom objects, and must be dot-qualified with the name of the article type to which they belong. This is shown in the following sample `package.xml` file:

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <fullName>articlefilemetadata</fullName>
    <apiAccessLevel>Unrestricted</apiAccessLevel>
    <types>
        <members>newarticle__kav.description__c</members>
        <name>CustomField</name>
    </types>
    <types>
        <members>newarticle__kav</members>
        <name>CustomObject</name>
    </types>
</Package>
```

Version

ArticleTypes are available in API version 19.0 and later.

Fields

Field Name	Field Type	Description
`articleTypeChannelDisplay`	articleTypeChannelDisplay	Represents the article-type templates used to display an article in the various channels. See "Article Type Templates" in the Salesforce online help.
`deploymentStatus`	DeploymentStatus (enumeration of type string)	A string which represents the deployment status of a custom object or field. Valid values are: • `InDevelopment` • `Deployed`
`description`	string	A description of the article type. Maximum of 1000 characters.
`fields`	CustomField[]	Represents one or more fields in the article type.
`gender`	Gender	Indicates the gender of the noun that represents the object. This is used for languages where words need different treatment depending on their gender.
`label`	string	Label that represents the object throughout the Salesforce user interface.
`pluralLabel`	string	Plural version of the `label` value.

Field Name	Field Type	Description
`startsWith`	StartsWith (enumeration of type string)	Indicates whether the noun starts with a vowel, consonant, or is a special character. This is used for languages where words need different treatment depending on the first character. Valid values are listed in StartsWith.

ArticleTypeChannelDisplay

Determines the article-type templates that are used to display an article in its channels. Unless otherwise noted, all fields are createable, filterable, and nillable.

Field Name	Field Type	Description
`articleTypeTemplates`	ArticleTypeTemplate on page 313[]	Indicates which article-type template applies in the specified channel.

ArticleTypeTemplate

Sets the article-type template for a specific channel. If not specified, the default article-type template applies.

Field Name	Field Type	Description
`channel`	string	Specifies the channel where the article-type template applies: • `AllChannels`: all the available channels. • `App`: the Articles tab in Salesforce Knowledge. • `Pkb`: the public knowledge base. • `Csp`: the Customer Portal. • `Prm`: the partner portal.
`page`	string	Represents the name of the custom Visualforce page used as a custom article-type template. Use this field when you select `Page` in the template field.
`template`	string	Indicates the article-type template used for the specified channel: • `Page`: custom Visualforce page. When specifying this value, you must also set the `page` field with the Visualforce page name. • `Tab`: display the sections you defined in the layout as tabs. • `Toc`: display the sections you defined in the layout as table of content.

Declarative Metadata Sample Definitions

A sample article type definition follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<CustomObject xmlns="http://soap.sforce.com/2006/04/metadata">
    <articleTypeChannelDisplay>
```

```
        <articleTypeTemplates>
            <channel>App</channel>
            <template>Tab</template>
        </articleTypeTemplates>
        <articleTypeTemplates>
            <channel>Prm</channel>
            <template>Tab</template>
        </articleTypeTemplates>
        <articleTypeTemplates>
            <channel>Csp</channel>
            <template>Tab</template>
        </articleTypeTemplates>
        <articleTypeTemplates>
            <channel>Pkb</channel>
            <template>Toc</template>
        </articleTypeTemplates>
    </articleTypeChannelDisplay>
    <deploymentStatus>Deployed</deploymentStatus>
    <description>Article type with custom fields</description>
    <fields>
        <fullName>description__c</fullName>
        <label>Description</label>
        <length>48</length>
        <type>Text</type>
    </fields>
    <label>newarticle</label>
    <pluralLabel>newarticles</pluralLabel>
</CustomObject>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ArticleType Layout

Represents the metadata associated with an article type page layout. Article type layouts determine which fields users can view and edit when entering data for an article. Article type layouts also determine which sections appear when users view articles.

ChannelLayout

Represents the metadata associated with a communication channel layout. Communication channel layouts let admins share article content inline into communication channels (for example, in email publishers, Experience Builder sites, or social media publishers). Admins can create a list of fields for an article type or record type that they want to share for each communication channel. You can customize the order of the fields.

ArticleType CustomField

Represents the metadata associated with an article type custom field. Use this metadata type to create, update, or delete article type custom field definitions.

SEE ALSO:

ArticleType Layout

ArticleType CustomField

ArticleType Layout

Represents the metadata associated with an article type page layout. Article type layouts determine which fields users can view and edit when entering data for an article. Article type layouts also determine which sections appear when users view articles.

The format of the article, for example whether layout sections display as subtabs or as a single page with links, is defined by the article-type template. Each article type has only one layout, but you can choose a different template for each of the article type's four channels. See Knowledge in *SOAP API Developer Guide*.

File Suffix and Directory Location

ArticleType layouts are stored in the `layouts` directory of the corresponding package directory. The prefix must match with the article type API name. The extension is `.layout`.

Version

ArticleType layouts are available in API version 19.0 and later.

Fields

Field Name	Field Type	Description
`layoutSections`	LayoutSection[]	The main sections of the layout containing the article fields. The order here determines the layout order.

LayoutSection

LayoutSection represents a section of an ArticleType layout.

Field Name	Field Type	Description
`customLabel`	boolean	Indicates if this section's label is custom or standard (built-in). Custom labels can be any text, but must be translated. Standard labels have a predefined set of valid values, for example 'System Information', which are automatically translated.
`label`	string	The label; either standard or custom, based on the `customLabel` flag.
`layoutColumns`	LayoutColumn[]	The columns of the layout, depending on the style. Salesforce Knowledge only supports one column in article type layouts.
`style`	LayoutSectionStyle (enumeration of type string)	The style of the layout. Salesforce Knowledge only supports the value `OneColumn`, which displays a one-column page.

LayoutColumn

LayoutColumn represents the items in a column within a layout section.

Field Name	Field Type	Description
`layoutItems`	LayoutItem[]	The individual items within a column (ordered from top to bottom).

LayoutItem

LayoutItem represents the valid values that define a layout item.

Field Name	Field Type	Description
`field`	string	The field name reference, for example `MyField__c`.

Declarative Metadata Sample Definition

The following is the definition of an ArticleType page layout:

```
<?xml version="1.0" encoding="UTF-8"?>
<Layout xmlns="http://soap.sforce.com/2006/04/metadata">
    <layoutSections>
        <customLabel>true</customLabel>
        <label>Description</label>
        <layoutColumns>
            <layoutItems>
                <field>description__c</field>
            </layoutItems>
            <layoutItems>
                <field>dateTime__c</field>
            </layoutItems>
        </layoutColumns>
        <style>OneColumn</style>
    </layoutSections>
    <layoutSections>
        <label>Data Sheet</label>
        <layoutColumns>
            <layoutItems>
                <field>file__c</field>
            </layoutItems>
        </layoutColumns>
        <style>OneColumn</style>
    </layoutSections>
</Layout>
```

SEE ALSO:

ArticleType

ArticleType CustomField

ChannelLayout

Represents the metadata associated with a communication channel layout. Communication channel layouts let admins share article content inline into communication channels (for example, in email publishers, Experience Builder sites, or social media publishers). Admins can create a list of fields for an article type or record type that they want to share for each communication channel. You can customize the order of the fields.

File Suffix and Directory Location

Channel layout components have the suffix `.channelLayout` and are stored in the `channelLayouts` folder of the corresponding package directory. The prefix must match with the article type API name. In Lightning Knowledge, the prefix must match the API name for the knowledge object.

Version

Channel layout components are available in API version 32.0 and later.

Fields

Field Name	Field Type	Description
`doesExcludeFieldLabels`	boolean	Indicates whether field labels are excluded from the field contents in the communication channels where this layout applies (`true`) or not (`false`). The default is `false`, meaning field labels are inserted. Available when Lightning Knowledge is enabled in API version 48.0 and later.
`doesExcludeFiles`	boolean	Indicates whether related files are left off emails (`true`) or attached to emails (`false`). The default is `false`, meaning related files are attached. Available when Lightning Knowledge is enabled in API version 48.0 and later.
`enabledChannels`	string[]	The communication channels where this layout applies. In API version 32.0 to 46.0, the only valid value is `Email`. When Lightning Knowledge is enabled in API version 47.0 and later, `Chat`, `Messaging`, and `Social` are added valid values.
`label`	string	Required. The label for this configuration.
`layoutItems`	ChannelLayoutItem on page 318[]	The article fields contained in the layout. The order here determines the field order.
`recordType`	string	The name of the record type that the channel layout applies to. The default is the primary record type. Available in API version 41.0 and later.

ChannelLayoutItem

Field Name	Field Type	Description
`field`	string	Required. Name of the field. The format is *`ArticleTypeName.FieldName`* or, in Lightning Knowledge, *`KnowledgeBaseName.FieldName`*.

Declarative Metadata Sample Definition

The following is an example of a ChannelLayout component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ChannelLayout xmlns="http://soap.sforce.com/2006/04/metadata">
    <label>Layout for Email</label>
    <layoutItems>
        <field>Knowledge.Question</field>
    </layoutItems>
    <layoutItems>
        <field>Knowledge.Answer</field>
    </layoutItems>
    <enabledChannels>Email</enabledChannels>
    <enabledChannels>Social</enabledChannels>
    <enabledChannels>Chat</enabledChannels>
    <doesExcludeFiles>false</doesExcludeFiles>
    <doesExcludeFieldLabels>true</doesExcludeFieldLabels>
</ChannelLayout>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>ChannelLayout</name>
    </types>
    <version>41.0</version>
</Package>
```

ArticleType CustomField

Represents the metadata associated with an article type custom field. Use this metadata type to create, update, or delete article type custom field definitions.

This type extends the Metadata metadata type and inherits its `fullName` field.

Always specify the full name whenever you create or update a custom field. For example, a custom field on a custom object:

```
MyArticleType__kav.MyCustomField__c
```

Declarative Metadata File Suffix and Directory Location

Custom fields are defined as part of the article type. ArticleType field names have a suffix of `__c` like other custom objects, and must be dot-qualified with the name of the article type to which they belong. See ArticleType for more information.

Retrieving Custom Fields on Custom or Standard Objects

When you retrieve a custom or standard object, you return everything associated with the object. However, you can also retrieve only the custom fields for an object by explicitly naming the object and fields in `package.xml`. The following definition in `package.xml` retrieves the files `objects/MyCustomObject__c.object`, `objects/Account.object__c.object`, and `objects/MyArticleType__kav.object`, each containing one custom field definition.

```
<types>
    <members>MyCustomObject__c.MyCustomField__c</members>
    <members>Account.MyCustomAccountField__c</members>
    <members>MyArticleType__kav.MyOtherCustomField__c</members>
    <name>CustomField</name>
</types>
```

Version

ArticleTypes custom fields are available in API version 19.0 and later.

Fields for ArticleType

Unless otherwise noted, all fields are createable, filterable, and nillable.

Note: If you create a knowledge validation rule, the errors always display at the top of the page, even if you add it beside the field. Therefore, write the errors descriptively so authors know how to satisfy the validation rule. For example, identify which field is causing the error. The Salesforce Classic user interface does not support field level error messages for articles.

Field Name	Field Type	Description
`defaultValue`	string	If specified, represents the default value of the field. This field was deprecated in API version 48.0.
`deleteConstraint`	Metadata Field Types (enumeration of type string)	Provides deletion options for lookup relationships. Valid values are: • `Cascade`—Deletes the lookup record as well as associated lookup fields. • `Restrict`—Prevents the record from being deleted if it's in a lookup relationship. • `SetNull`—This is the default. If the lookup record is deleted, the lookup field is cleared. For more information on lookup relationships, see "Object Relationships" in Salesforce Help.
`description`	string	Description of the field.

Field Name	Field Type	Description
formula	string	If specified, represents a formula on the field.
formulaTreatBlankAs	Metadata Field Types (enumeration of type string)	Indicates how to treat blanks in a formula. Valid values are: BlankAsBlank and BlankAsZero.
fullName	string	Inherited from Metadata, this field is defined in the WSDL for this metadata type. It must be specified when creating, updating, or deleting. See createMetadata() to see an example of this field specified for a call. This value cannot be null.
inlineHelpText	string	Represents the content of field-level help. For more information, see "Define Field-Level Help" in Salesforce Help.
label	string	Label for the field. You cannot update the label for standard fields in Article Type such as Title, UrlName, Summary, etc.
length	int	Length of the field.
picklist	Picklist (Including Dependent Picklist)	(**Deprecated.** Use this field in API version 37.0 and earlier only. In later versions, use valueSet instead.) If specified, the field is a picklist, and this field enumerates the picklist values and labels.
referenceTo	string	If specified, indicates a reference this field has to another object.
relationshipLabel	string	Label for the relationship.
relationshipName	string	If specified, indicates the value for one-to-many relationships. For example, in the object MyObject that had a relationship to YourObject, the relationship name might be YourObjects.
required	boolean	Indicates whether the field requires a value on creation (true) or not (false).
type	FieldType	Required. Indicates the field type for the field. Valid values are: • Checkbox available in version 30.0 and later • Currency • ArticleCurrency • Date • DateTime • Email • File • Formula • Html • Lookup • Number

Field Name	Field Type	Description
		• Percent • Phone • Picklist • DependentPicklist • MultiselectPicklist • Text • TextArea • LongTextArea • URL
visibleLines	int	Indicates the number of lines displayed for the field.

Declarative Metadata Sample Definition

```
<?xml version="1.0" encoding="UTF-8"?>
<CustomObject xmlns="http://soap.sforce.com/2006/04/metadata">
    <fields>
        <fullName>Comments__c</fullName>
        <description>add your comments about this object here</description>
        <label>Comments</label>
        <length>32000</length>
        <type>LongTextArea</type>
        <visibleLines>30</visibleLines>
    </fields>
</CustomObject>
```

SEE ALSO:

ArticleType

ArticleType Layout

ApexClass

Represents an Apex class. An Apex class is a template or blueprint from which Apex objects are created. Classes consist of other classes, user-defined methods, variables, exception types, and static initialization code.

For more information, see the Lightning Platform Apex Code Developer's Guide. This type extends the MetadataWithContent metadata type and inherits its `content` and `fullName` fields.

Note: By default, you can't deploy updates to an Apex class if there are one or more active jobs for that class. To deploy updates in this case, do one of the following.

- Cancel Apex jobs before deploying changes to Apex code. Reschedule the jobs after the deployment.
- Enable deployments with Apex jobs in the Salesforce user interface in the Deployment Settings page.

Supported Calls

All Metadata API calls except CRUD-Based Calls, which prevents deployment outside of proper deployment lifecycle and test-execution constraints.

Declarative Metadata File Suffix and Directory Location

The file suffix is `.cls` for the class file. The accompanying metadata file is named *`ClassName`*`.cls-meta.xml`.

Apex classes are stored in the `classes` folder in the corresponding package directory.

Version

Apex classes are available in API version 10.0 and later.

Fields

This metadata type contains the following fields:

Field Name	Field Type	Description
`apiVersion`	double	The API version for this class. Every class has an API version specified at creation.
`content`	base64	The Apex class definition. Base 64-encoded binary data. Before making an API call, client applications must encode the binary attachment data as base64. Upon receiving a response, client applications must decode the base64 data to binary. This conversion is handled for you by a SOAP client. This field is inherited from the MetadataWithContent component.
`fullName`	string	The Apex class name. The name can only contain characters, letters, and the underscore (_) character, must start with a letter, and can't end with an underscore or contain two consecutive underscore characters. This field is inherited from the Metadata component.
`packageVersions`	PackageVersion[]	The list of installed managed package versions that are referenced by this Apex class. For more information about managed packages, see Second-Generation Managed Packages in the *Salesforce DX Developer Guide*. This field is available in API version 16.0 and later.
`status`	ApexCodeUnitStatus (enumeration of type string)	The status of the Apex class. The following string values are valid: • `Active` - The class is active. • `Deleted` - The class is marked for deletion. This value is useful for managed packages, because it allows a class to be deleted when a managed package is updated. ApexCodeUnitStatus includes an `Inactive` option, but it's only supported for ApexTrigger; it isn't supported for ApexClass.

PackageVersion

PackageVersion identifies a version of a managed package. A package version is a number that identifies the set of components included in a package. The version number has the format *majorNumber.minorNumber.patchNumber* (for example, 2.1.3). The major and minor numbers increase to a chosen value during every major release. The *patchNumber* is generated and updated only for a patch release. It's available in API version 16.0 and later.

Field Name	Field Type	Description
namespace	string	Required. In a packaging context, a namespace prefix is a one to 15-character alphanumeric identifier that distinguishes your package and its contents from packages of other developers on AppExchange. Namespace prefixes are case-insensitive. For example, ABC and abc aren't recognized as unique. Your namespace prefix must be globally unique across all Salesforce orgs. Salesforce automatically prepends your namespace prefix, followed by two underscores ("__"), to all unique component names in your Salesforce organization. A unique package component is one that requires a name that no other component has within Salesforce, such as custom objects, custom fields, custom links, s-controls, and validation rules. For more information about namespaces, see Create and Register Your Namespace in the *Second-Generation Managed Packaging Developer Guide*.
majorNumber	int	Required. The major number of the package version. A package version number has a *majorNumber.minorNumber* format.
minorNumber	int	Required. The minor number of the package version. A package version number has a *majorNumber.minorNumber* format.

Declarative Metadata Sample Definition

The following sample creates the `MyhelloWorld.cls` class, and the corresponding `MyHelloWorld.cls-meta.xml` metadata file.

`MyHelloWorld.cls` file:

```
public class MyHelloWorld {
// This method updates the Hello field on a list
// of accounts.
public static void addHelloWorld(Account[] accs){
 for (Account a:accs){
  if (a.Hello__c != 'World')
  a.Hello__c = 'World';
  }
 }
}
```

`MyHelloWorld.cls-meta.xml`:

```
<?xml version="1.0" encoding="UTF-8"?>
<ApexClass xmlns="http://soap.sforce.com/2006/04/metadata">
    <apiVersion>64.0</apiVersion>
</ApexClass>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SEE ALSO:

ApexTrigger

ApexComponent

Represents a Visualforce component.

For more information, see Visualforce in Salesforce Help and StaticResource: MetadataWithContent on page 2185

Declarative Metadata File Suffix and Directory Location

The file suffix is `.component` for the page file. The accompanying metadata file is named *`ComponentName`*`-meta.xml`.

Visualforce components are stored in the `components` folder in the corresponding package directory.

Version

Visualforce components are available in API version 12.0 and later.

Fields

This metadata type contains the following fields:

Field Name	Field Type	Description
`apiVersion`	double	The API version for this Visualforce component. Every component has an API version specified at creation. This field is available in API version 16.0 and later.
`content`	base64Binary	The component content. Base 64-encoded binary data. Before making an API call, client applications must encode the binary attachment data as base64. Upon receiving a response, client applications must decode the base64 data to binary. This conversion is handled for you by a SOAP client. This field is inherited from the MetadataWithContent component.
`description`	string	A description of what the component does.
`fullName`	string	The component developer name used as a unique identifier for API access. The `fullName` can contain only underscores and alphanumeric characters. It must be unique, begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. This field is inherited from the Metadata component.
`label`	string	Required. The label for this component.

Field Name	Field Type	Description
packageVersions	PackageVersion[]	The list of installed managed package versions that are referenced by this Visualforce component. Package components and Visualforce custom component are distinct concepts. A package is comprised of many elements, such as custom objects, Apex classes and triggers, and custom pages and components. For more information about managed packages, see Second-Generation Managed Packages in the *Salesforce DX Developer Guide*. This field is available in API version 16.0 and later.

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SEE ALSO:

ApexPage

ApexEmailNotifications

The ApexEmailNotifications type allows you to define users and email addresses that receive email for unhandled Apex errors. Flow errors can also use this metadata type.

Declarative Metadata File Suffix and Directory Location

The component filename is `apexEmailNotifications.notifications`. The Apex email notification file is stored in the `apexEmailNotifications` folder in the corresponding package directory.

Version

ApexEmailNotifications components are available in API version 49.0 and later.

Fields

Field Name	Field Type	Description
apexEmailNotification	ApexEmailNotification	A specific Apex email notification. You can specify multiple notifications.

ApexEmailNotification

Represents an Apex email notification.

Note: Each ApexEmailNotification can contain an email or a user but not both.

Field Name	Field Type	Description
`email`	string	The external email address to which the notification is sent. Mutually exclusive with the `user` field.
`user`	string	The username of the Salesforce user to be notified. Mutually exclusive with the `email` field.

Usage

Deploying ApexEmailNotifications deletes all previous notifications in the org. For example, consider two notifications, test1@example.com and test2@example.com, that are deployed in an org. When the following `apexEmailNotifications.notifications` is deployed, test1@example.com is deleted, because it's not in the deployed list.

```
<?xml version="1.0" encoding="UTF-8"?>
<ApexEmailNotifications xmlns="http://soap.sforce.com/2006/04/metadata">
    <apexEmailNotification>
        <email>test2@example.com</email>
    </apexEmailNotification>
</ApexEmailNotifications>
```

Note: The ApexEmailNotifications metadata type isn't supported in `destructiveChanges.xml`. To delete specific ApexEmailNotification items, deploy a new ApexEmailNotifications without those items. To delete all Apex email notifications in an org, deploy an empty list of ApexEmailNotifications.

Declarative Metadata Sample Definition

To deploy Apex email notifications, you can specify either the exact file name or use a wildcard in `package.xml`.

This example specifies the exact file name in `package.xml`.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>apexEmailNotifications</members>
        <name>ApexEmailNotifications</name>
    </types>
    <version>49.0</version>
</Package>
```

This example uses a wildcard in `package.xml`.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>ApexEmailNotifications</name>
    </types>
    <version>49.0</version>
</Package>
```

This sample deploys an Apex email notification that notifies a Salesforce user in the org.

```
<?xml version="1.0" encoding="UTF-8"?>
<ApexEmailNotifications xmlns="http://soap.sforce.com/2006/04/metadata">
    <apexEmailNotification>
        <user>user1@example.com</user>
    </apexEmailNotification>
</ApexEmailNotifications>
```

This sample deploys an Apex email notification that notifies an external email address.

```
<?xml version="1.0" encoding="UTF-8"?>
<ApexEmailNotifications xmlns="http://soap.sforce.com/2006/04/metadata">
    <apexEmailNotification>
        <email>test@example.com</email>
    </apexEmailNotification>
</ApexEmailNotifications>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ApexPage

Represents a Visualforce page.

For more information, see *Visualforce* in Salesforce Help. This type extends the MetadataWithContent metadata type and inherits its `content` and `fullName` fields.

Declarative Metadata File Suffix and Directory Location

The file suffix is `.page` for the page file. The accompanying metadata file is named *`PageName`*`-meta.xml`.

Visualforce pages are stored in the `pages` folder in the corresponding package directory.

Version

Visualforce pages are available in API version 11.0 and later.

Fields

This metadata type contains the following fields:

Field Name	Field Type	Description
`apiVersion`	double	Required. The API version for this page. Every page has an API version specified at creation. This field is available in API version 15.0 and later. If you set this field to a number lower than 15.0, it's changed to 15.0.

Field Name	Field Type	Description
`content`	base64Binary	The page content. Base 64-encoded binary data. Before making an API call, client applications must encode the binary attachment data as base64. Upon receiving a response, client applications must decode the base64 data to binary. This conversion is handled for you by a SOAP client. This field is inherited from the MetadataWithContent component.
`description`	string	A description of what the page does.
`fullName`	string	The page developer name used as a unique identifier for API access. The `fullName` can contain only underscores and alphanumeric characters. It must be unique, begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. This field is inherited from the Metadata component.
`availableInTouch`	boolean	Indicates if Visualforce tabs associated with the Visualforce page can be used in the Salesforce mobile app. (Use of this field for Salesforce Touch is deprecated.). This field is available in API version 27.0 and later. Standard object tabs that are overridden with a Visualforce page aren't supported in the Salesforce mobile app, even if you set this field for the page. The default page for the object is displayed instead of the Visualforce page.
`confirmationTokenRequired`	boolean	Indicates whether `GET` requests for the page require a CSRF confirmation token. This field is available in API version 28.0 and later. If you change this field's value from `false` to `true`, links to the page require a CSRF token to be added to them, or the page is inaccessible.
`label`	string	Required. The label for this page.
`packageVersions`	PackageVersion[]	The list of installed managed package versions that are referenced by this Visualforce page. For more information about managed packages, see Second-Generation Managed Packages in the *Salesforce DX Developer Guide*. This field is available in API version 16.0 and later.

Declarative Metadata Sample Definition

The following sample creates the `MyPage.page` page, and the corresponding `MyPage.page-meta.xml` metadata file.

`SampleApexPage.page` file:

```
<apex:page>
<h1>Congratulations</h1>
This is your new Page.
</apex:page>
```

`SampleApexPage.page-meta.xml`:

```
<?xml version="1.0" encoding="UTF-8"?>
<ApexPage xmlns="http://soap.sforce.com/2006/04/metadata">
    <description>This is a sample Visualforce page.</description>
    <label>SampleApexPage</label>
</ApexPage>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SEE ALSO:

ApexComponent

ApexTestSuite

Represents a suite of Apex test classes to include in a test run.

File Suffix and Directory Location

ApexTestSuite components have the suffix `.testSuite` and are stored in the `testSuites` folder.

Version

ApexTestSuite components are available in API version 38.0 and later.

Fields

Field Name	Field Type	Description
`testClassName`	string[]	A list of Apex test classes, specified by name, to include in this test suite.

Declarative Metadata Sample Definition

To include namespaced tests in an Apex test suite, specify each namespace individually. Local Apex tests consist of all tests in the org that don't originate from managed packages.

```
<?xml version="1.0" encoding="UTF-8"?>
<ApexTestSuite xmlns="http://soap.sforce.com/2006/04/metadata">
   <testClassName>LocalTestClass</testClassName>
      <!-- LocalTestClass adds the test class named LocalTestClass. -->
   <testClassName>A*Class</testClassName>
      <!-- A*Class adds AClass, AnotherClass, AwesomeClass, and so on. -->
```

```
    <testClassName>Namespace1.NamespacedTestClass</testClassName>
    <testClassName>*</testClassName> <!-- Adds all local tests. -->
    <testClassName>Namespace1.*</testClassName> <!-- Adds all tests in Namespace1. -->
    <testClassName>Namespace2.*</testClassName> <!-- Adds all tests in Namespace2. -->
</ApexTestSuite>
```

These syntaxes are supported in `package.xml`. If the test classes in your suites are already present in the target org, you can omit the `ApexClass` type in `package.xml`.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>ApexClass</name>
    </types>
    <types>
        <members>*</members>
        <name>ApexTestSuite</name>
    </types>
    <version>38.0</version>
</Package>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>ApexClass</name>
    </types>
    <types>
        <members>Suite1</members>
        <members>Suite2</members>
        <name>ApexTestSuite</name>
    </types>
    <version>38.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ApexTrigger

Represents an Apex trigger. A trigger is Apex code that executes before or after specific data manipulation language (DML) events occur, such as before object records are inserted into the database, or after records have been deleted.

For more information, see "Manage Apex Triggers" in Salesforce Help. This type extends the MetadataWithContent metadata type and inherits its `content` and `fullName` fields.

Supported Calls

All Metadata API calls except CRUD-Based Calls, which prevents deployment outside of proper deployment lifecycle and test-execution constraints.

Declarative Metadata File Suffix and Directory Location

The file suffix is `.trigger` for the trigger file. The accompanying metadata file is named *`TriggerName`*`-meta.xml`.

Apex triggers are stored in the `triggers` folder in the corresponding package directory.

Version

Triggers are available in API version 10.0 and later.

Fields

This metadata type contains the following fields:

Field Name	Field Type	Description
`apiVersion`	double	Required. The API version for this trigger. Every trigger has an API version specified at creation.
`content`	base64	The Apex trigger definition. This field is inherited from the MetadataWithContent component.
`fullName`	string	The Apex trigger name. The name can only contain characters, letters, and the underscore (_) character, must start with a letter, and can't end with an underscore or contain two consecutive underscore characters. This field is inherited from the Metadata component.
`packageVersions`	PackageVersion[]	The list of installed managed package versions that are referenced by this Apex trigger. For more information about managed packages, see the Second-Generation Managed Packaging Developer Guide. This field is available in API version 16.0 and later.
`status`	ApexCodeUnitStatus (enumeration of type string)	Required. The status of the Apex trigger. The following string values are valid: • `Active` - The trigger is active. • `Inactive` - The trigger is inactive, but not deleted. • `Deleted` - The trigger is marked for deletion. Useful for managed packages, because it allows a trigger to be deleted when a managed package is updated.

Declarative Metadata Sample Definition

The following sample creates the `MyhelloWorld.trigger` trigger, and the corresponding `MyHelloWorld.trigger-meta.xml` metadata file.

`MyHelloWorld.trigger` file:

```
trigger helloWorldAccountTrigger on Account (before insert) {

  Account[] accs = Trigger.new;

   MyHelloWorld.addHelloWorld(accs);
}
```

`MyHelloWorld.trigger-meta.xml`:

```
<?xml version="1.0" encoding="UTF-8"?>
<ApexTrigger xmlns="http://soap.sforce.com/2006/04/metadata">
    <apiVersion>64.0</apiVersion>
</ApexTrigger>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SEE ALSO:

ApexClass

AppMenu

Represents the app menu or the Salesforce mobile navigation menu. Reserved for future use.

AppointmentAssignmentPolicy

Represents the information about a resource assignment rule. This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

AppointmentAssignmentPolicy components have the suffix `.policy` and are stored in the `appointmentSchedulingPolicies` folder.

Version

AppointmentSchedulingPolicy components are available in API version 53.0 and later.

Fields

Field Name	Field Type	Description
masterLabel	string	Required. The label for the appointment assignment policy.
policyApplicableDuration	string	Required. The frequency at which the utilization of service resources is calculated. Valid values are: • Monthly • ParameterBased • Weekly
policyType	string	Required. The type of appointment assignment policy. Valid value is: • loadBalancing
utilizationFactor	string	Required. Specifies the count type for the resource utilization. Valid values are: • NumberOfAppointments • TotalAppointmentDuration

Declarative Metadata Sample Definition

The following is an example of an appointmentAssignmentPolicy component.

```
<?xml version="1.0" encoding="UTF-8"?>
<AppointmentAssignmentPolicy xmlns="http://soap.sforce.com/2006/04/metadata">
   <masterLabel>loadBalancing Assignment Policy</masterLabel>
   <policyType>loadBalancing</policyType>
   <policyApplicableDuration>Weekly</policyApplicableDuration>
   <utilizationFactor>TotalAppointmentDuration</utilizationFactor>
</AppointmentAssignmentPolicy>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
 <Package xmlns="http://soap.sforce.com/2006/04/metadata">
   <types>
     <members>sample</members>
     <name>AppointmentAssignmentPolicy</name>
   </types>
   <version>53.0</version>
 </Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

AppointmentSchedulingPolicy

Represents a set of rules for scheduling appointments using Lightning Scheduler. This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

AppointmentSchedulingPolicy components have the suffix `.policy` and are stored in the `appointmentSchedulingPolicies` folder.

Version

AppointmentSchedulingPolicy components are available in API version 47.0 and later.

Special Access Rules

You must have the ViewSetup and CustomizeApplication user permissions to access the AppointmentSchedulingPolicy type.

Fields

Field Name	Field Type	Description
`appointmentAssignmentPolicy`	string	The name of the appointment assignment policy. This field is available in API version 53.0 and later.
`appointmentStartTimeInterval`	picklist	Required. The proposed time interval in minutes between appointment start times. For example, if you set the interval to `15`, appointments can then begin at the top of the hour and at 15-minute intervals thereafter (10:00 AM, 10:15 AM, 10:30 AM, and so on). Valid values are: • `5` • `10` • `15` • `20` • `30` • `45` • `60` • `90` • `120` • `150` • `180` • `240`

Field Name	Field Type	Description
		• 300 • 360 • 420 • 480
extCalEventHandler	lookup	Required. The API name of the custom Apex class that checks service resources' external calendar events and returns the time slots where service resources are already booked. Available in API version 50.0 and later.
isSvcTerritoryMemberShiftUsed	boolean	Required. Indicates whether to consider shifts of service territory members when determining the availability of service resources for appointments (true) or not (false). This field is available in API version 54.0 and later.
isSvcTerrOpHoursWithShiftsUsed	boolean	Required. Indicates whether to consider the intersection of shifts and service territory operating hours when determining the availability of service resources for appointments (true) or not (false). This field is available in API version 54.0 and later.
masterLabel	string	Required. The label for the appointment scheduling policy.
shouldCheckExternalCalendar	boolean	Required. Indicates whether to check the external calendar for resource availability (true) or not (false). This field is available in API version 53.0 and later.
shouldConsiderCalendarEvents	boolean	Required. Indicates whether to consider events on the Salesforce calendar to determine the availability of service resources to be assigned to appointments (true) or not (false).
shouldEnforceExcludedResource	boolean	Required. Indicates whether this appointment scheduling policy prevents excluded service resources from being assigned to appointments (true) or not (false).
shouldEnforceRequiredResource	boolean	Required. Indicates whether this appointment scheduling policy allows only required service resources to be assigned to appointments (true) or not (false).
shouldMatchSkill	boolean	Required. Indicates whether this appointment scheduling policy allows only required service resources who have certain skills to be assigned to appointments (true) or not (false).
shouldMatchSkillLevel	boolean	Required. Indicates whether this appointment scheduling policy allows only required service resources who have certain skills and skill levels to be assigned to appointments (true) or not (false).
shouldRespectVisitingHours	boolean	Required. Indicates whether this appointment scheduling policy prevents users from scheduling appointments outside of an account's visiting hours (true) or not (false).

Field Name	Field Type	Description
shouldUsePrimaryMembers	boolean	Required. Indicates whether this appointment scheduling policy allows only service resources who are primary members of a service territory to be assigned to appointments (true) or not (false).
shouldUseSecondaryMembers	boolean	Required. Indicates whether this appointment scheduling policy allows service resources who are secondary members of a service territory to be assigned to appointments (true) or not (false).

Declarative Metadata Sample Definition

The following is an example of an appointmentSchedulingPolicy component.

```
<?xml version="1.0" encoding="UTF-8"?>
<AppointmentSchedulingPolicy xmlns="http://soap.sforce.com/2006/04/metadata">
  <appointmentAssignmentPolicy>ResourceAssignmentRule1</appointmentAssignmentPolicy>
  <appointmentStartTimeInterval>15</appointmentStartTimeInterval>
  <masterLabel>Default Appointment Scheduling Policy</masterLabel>
  <shouldCheckExternalCalendar>true</shouldCheckExternalCalendar>
  <shouldConsiderCalendarEvents>true</shouldConsiderCalendarEvents>
  <shouldEnforceExcludedResource>true</shouldEnforceExcludedResource>
  <shouldEnforceRequiredResource>true</shouldEnforceRequiredResource>
  <shouldMatchSkill>true</shouldMatchSkill>
  <shouldMatchSkillLevel>false</shouldMatchSkillLevel>
  <shouldRespectVisitingHours>true</shouldRespectVisitingHours>
  <shouldUsePrimaryMembers>true</shouldUsePrimaryMembers>
  <shouldUseSecondaryMembers>true</shouldUseSecondaryMembers>
</AppointmentSchedulingPolicy>
```

The following is an example package.xml that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
 <Package xmlns="http://soap.sforce.com/2006/04/metadata">
   <types>
     <members>sample</members>
     <name>AppointmentSchedulingPolicy</name>
   </types>
   <version>47.0</version>
 </Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character * (asterisk) in the package.xml manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ApprovalProcess

Represents the metadata associated with an approval process. An approval process automates how records are approved in Salesforce. An approval process specifies each step of approval, including who to request approval from and what to do at each point of the process.

This type extends the Metadata metadata type and inherits its `fullName` field.

Note:

- To use approval processes on Salesforce Knowledge articles with the Metadata API, the article type must be deployed. For article version (_kav) in approval processes, the supported action types are: Knowledge Action, Email Alert, Field Update, and Outbound Message.
- Send actions and approval processes for email drafts aren't supported in the Metadata API.
- The metadata doesn't include the order of active approval processes. Sometimes you have to reorder the approval processes in the destination org after deployment.

File Suffix and Directory Location

ApprovalProcess components have the suffix `.approvalProcess` and are stored in the `approvalProcesses` folder.

Version

ApprovalProcess components are available in API version 28.0 and later.

Fields

Field Name	Field Type	Description
`active`	boolean	Required. Whether the approval process is active. After an approval process is activated, you can't add, delete, or change the order of the steps or change its reject or skip behavior, even if the process is inactive.
`allowRecall`	boolean	Whether to allow submitters to recall approval requests. If set to `false`, only administrators can recall approval requests.
`allowedSubmitters`	ApprovalSubmitter[]	Required. An array of users who are allowed to submit records for approval.
`approvalPageFields`	ApprovalPageField	Specifies which fields to display on the approval page, where the approver goes to approve or reject the record. By default, the approval page displays the following: • `Name` field • `Owner` field (except for child objects) If you enable notifications in the Salesforce mobile app, keep in mind that approvers can view this list of fields on a mobile

Field Name	Field Type	Description
		device. Select only the fields necessary for users to decide whether to approve or reject records.
`approvalStep`	ApprovalStep[]	An array of approval step definitions.
`description`	string	Describes the approval process.
`emailTemplate`	string	Specifies which Classic email template to use for approval requests. If not specified, the default email template is used. Lightning email templates aren't packageable. We recommend using a Classic email template. When an approval process assigns an approval request to a user, Salesforce sends the user an approval request email.
`enableMobileDeviceAccess`	boolean	Whether users can access an external version of the approval page from any browser, including browsers on mobile devices, without logging in to Salesforce. Corresponds to `Security Settings` in the user interface. If set to `true`, approval steps can't have approvers of `type adhoc`. If set to `false`, approvers must log in to Salesforce to access the approval page.
`entryCriteria`	ApprovalEntryCriteria	Determines which records can enter the approval process. Exclude this field to allow all records to enter the approval process. When you deploy an approval process with no entry criteria to overwrite an existing approval process with entry criteria, then the entry criteria from the existing process are applied to the deployed process.
`finalApprovalActions`	ApprovalAction	Specifies which workflow actions to execute when all required approvals have been given for a record.
`finalApprovalRecordLock`	boolean	Whether to keep the record locked after it receives all necessary approvals. Default: `false`.
`finalRejectionActions`	ApprovalAction	Specifies which workflow actions to execute after a record enters the final rejection state.
`finalRejectionRecordLock`	boolean	Whether to keep the record locked after it's finally rejected. Default: `false`.
`initialSubmissionActions`	ApprovalAction	Specifies which workflow actions to execute when a record is initially submitted for approval.
`label`	string	Required. Name of the approval process.

Field Name	Field Type	Description
`nextAutomatedApprover`	NextAutomatedApprover	Specifies a standard or custom user hierarchy field that can be used to automatically assign the approver for an approval step. If you exclude this field, then no approval step can use a user hierarchy field to automatically assign the approver.
`postTemplate`	string	Post template to use for Approvals in Chatter. Chatter post approval notifications are only available for approval processes associated with an object that has been enabled for feed tracking.
`recallActions`	ApprovalAction	Specifies which workflow actions to execute when a pending approval request is withdrawn.
`recordEditability`	RecordEditabilityType (enumeration of type string)	Specifies which users can edit records that are pending approval. When a record is submitted for approval, it's automatically locked to prevent other users from editing it during the approval process. Valid values are: • `AdminOnly`—Records pending approval can be edited by: – Users with the "Modify All Data" permission – Users with the "Modify All Records" object-level permission for the given object • `AdminOrCurrentApprover`—Records pending approval can be edited by: – Users with the "Modify All Data" permission – Users with the "Modify All Records" object-level permission for the given object – The assigned approver, who must have edit access to the record through user permissions and the organization-wide sharing defaults for the given object
`showApprovalHistory`	boolean	Whether to add the Approval History related list to the approval page, which is where the approver can view the approval request details and approve or reject the record. The Approval History related list tracks a record through the approval process. If you also want to add the Approval History related list to record detail and edit pages, use the Salesforce user interface to customize the page layouts for the given object.

ApprovalSubmitter

Represents a user or set of users who can submit records for approval.

Field Name	Field Type	Description

`submitter`	string	Identifies a specific user or set of users who can submit records for approval. This field is required, except when the following types are specified and the `submitter` field is ignored:
		• `owner`
		• `creator`
		• `allInternalUsers`
		Example:
		`<allowedSubmitters>`
		`  <type>allInternalUsers</type>`
		`</allowedSubmitters>`
		`<allowedSubmitters>`
		`  <submitter>myGroup</submitter>`
		`  <type>group</type>`
		`</allowedSubmitters>`

`type`	ProcessSubmitterType (enumeration of type string)	Required. Type of user or set of users who can submit records for approval. Valid values are: • `group` • `role` • `user` • `roleSubordinates` • `roleSubordinatesInternal` • `owner` • `creator` • `partnerUser` • `customerPortalUser` • `portalRole` • `portalRoleSubordinates` • `allInternalUsers`—all Salesforce users in the organization

ApprovalPageField

Represents the selection of fields to display on the approval page, where an approver can view the approval request details and approve or reject the record.

Field Name	Field Type	Description
`field`	string[]	An array of fields that are displayed on the page for the approver to approve or reject the record.

ApprovalStep

Represents a step in the approval process. Approval steps define the chain of approval for a particular approval process. Each step determines which records can advance to that step, who to assign approval requests to, and whether to let each approver's delegate respond to the requests. The first step specifies what to do if a record doesn't advance to that step. Later steps specify what happens if an approver rejects the request.

Note:

- The order of the `ApprovalStep` entries in the approval process definition determines the order in which the approval steps are executed.
- After an approval process is activated, you can't add, delete, or change the order of the steps or change its reject or skip behavior, even if the process is inactive.
- Each approval process supports up to 30 steps.

Field Name	Field Type	Description
`allowDelegate`	boolean	Whether to allow delegated approvers in this step of the approval process. A delegated approver is a user appointed by an assigned approver as an alternate for approval requests.
`approvalActions`	ApprovalAction	Specifies which workflow actions to execute when a record is approved in this step of the approval process.
`assignedApprover`	ApprovalStepApprover	Specifies the assigned approvers for this step of the approval process.
`description`	string	Describes the approval step.
`entryCriteria`	ApprovalEntryCriteria	Determines which records can enter this step of the approval process.
`ifCriteriaNotMet`	StepCriteriaNotMetType (enumeration of type string)	Specifies what to do for records that don't meet the entry criteria. Valid values are: • `ApproveRecord`—Approve the request and execute all final approval actions. • `RejectRecord`—Reject the request and execute all final rejection actions. This option is available only for the first step in the approval process. • `GotoNextStep`—Skip to the next approval step. If you select this option for the first approval step, and a record doesn't meet the entry criteria for any other step, the record is rejected.
`label`	string	Required. Name of the approval step.
`name`	string	Required. Unique name of the approval step. It must contain only underscores and alphanumeric characters, begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. The requirement for uniqueness is only within the specific approval process.

Field Name	Field Type	Description
`rejectBehavior`	ApprovalStepRejectBehavior	Required, except for the first step in the approval process. Specifies what happens if the approver rejects the request during this approval step, unless it's the first step in the approval process. If the approver rejects the request in the first step in the approval process, the reject behavior is determined by the `finalRejectionActions`.
`rejectionActions`	ApprovalAction	Specifies which workflow actions to execute when a record is rejected in this step of the approval process.

ApprovalAction

Represents the actions that occur as a result of an approval process.

Field Name	Field Type	Description
`action`	WorkflowActionReference[]	An array of workflow actions to execute.

ApprovalStepApprover

Represents the assigned approvers for an approval step. Each step supports up to 25 approvers.

Field Name	Field Type	Description
`approver`	Approver[]	An array of assigned approvers for this step of the approval process.
`whenMultipleApprovers`	RoutingType (enumeration of type string)	Specifies how to handle approval or rejection when multiple approvers are assigned to the step. Valid values are: • `Unanimous`—(Default) Require unanimous approval from all approvers for this step. If any of the approvers reject the request, the approval request for this step is rejected. • `FirstResponse`—Approve or reject based on the first response.

Approver

Represents an assigned approver for an approval step. Check out Considerations for Setting Approvers in Salesforce Help.

Field Name	Field Type	Description
`name`	string	Identifies an assigned approver. This field is required, except when the `type` is one of the following and the `name` is ignored: • `adhoc` • `userHierarchyField`

Field Name	Field Type	Description
`type`	NextOwnerType (enumeration of type string)	Combined with the specified `name`, `type` identifies an assigned approver. Valid values are: • `adhoc`—The approver for the step must be selected manually. For the first step, the submitter selects the approver. For the second and later steps, the approver for the previous step selects the approver. For this value, exclude the `name` field. • `user`—A user in your organization. For this value, enter a username for the `name` field. • `userHierarchyField`—A user specified in a standard or custom user hierarchy field, such as the standard `Manager` field. For this value, exclude the `name` field. The user hierarchy field must be defined in the `nextAutomatedApprovers` for the approval process. • `relatedUserField`—A user specified in a user lookup field on the submitted record, such as the `Last Modified By` field. For this value, enter the name of the user lookup field for the `name` field. • `queue`—Automatically assign to a queue. For this value, enter the name of the queue for the `name` field.

ApprovalEntryCriteria

Represents the criteria that records must meet to enter the approval process or an approval step. Specify either filter criteria or a formula, but not both.

Field Name	Field Type	Description
`booleanFilter`	string	Filter logic for `criteriaItems`. Exclude this field if you enter a `formula`.
`criteriaItems`	FilterItem[]	Filter criteria that a record must meet to enter the approval process or approval step. Approval processes don't support `valueField` entries in filter criteria.
`formula`	string	Formula that must evaluate to true for a record to enter the approval process or approval step.

ApprovalStepRejectBehavior

Represents what happens if the approver rejects the request during this approval step, unless it's the first step in the approval process. For the first step in the approval process, the reject behavior is determined by the approval process's final rejection actions.

Field Name	Field Type	Description
`type`	StepRejectBehaviorType (enumeration of type string)	Not allowed in the first step of the approval process. Valid values are: • `RejectRequest`—Rejects the request even if previous steps were approved. Salesforce performs all rejection actions specified for this step and all final rejection actions.

Field Name	Field Type	Description
		• `BackToPrevious`—Rejects the request, and returns the approval request to the previous approver. Salesforce performs all rejection actions specified for this step.

NextAutomatedApprover

Represents the user hierarchy field to use as the next automated approver for the approval process. If defined, the user specified in the hierarchy field can be automatically assigned as the approver in one or more approval steps.

Field Name	Field Type	Description
`useApproverFieldOfRecordOwner`	boolean	Required. Whether the first executed approval step uses the specified `userHierarchyField` in the record owner's user record—instead of the submitter's user record—as the approver. All remaining steps use the specified `userHierarchyField` in the user record of the preceding step's approver.
`userHierarchyField`	string	Required. Standard or custom user hierarchy field whose value specifies which user to assign as the approver. For example, the standard `Manager` hierarchy field can be used to assign approvers for employee PTO (paid time off) requests.

Declarative Metadata Sample Definition

The following is an example of an ApprovalProcess component:

```
<?xml version="1.0" encoding="UTF-8"?>
<ApprovalProcess xmlns="http://soap.sforce.com/2006/04/metadata">
    <active>false</active>
    <allowRecall>false</allowRecall>
    <allowedSubmitters>
        <type>owner</type>
    </allowedSubmitters>
    <allowedSubmitters>
        <submitter>USSalesRep</submitter>
        <type>role</type>
    </allowedSubmitters>
    <allowedSubmitters>
        <submitter>MarketingGroup</submitter>
        <type>group</type>
    </allowedSubmitters>
    <allowedSubmitters>
        <submitter>kcooper@example.com</submitter>
        <type>user</type>
    </allowedSubmitters>
    <approvalPageFields>
        <field>Name</field>
```

```
        <field>Owner</field>
        <field>MyLeadCustomField__c</field>
        <field>Address</field>
    </approvalPageFields>
    <approvalStep>
        <allowDelegate>false</allowDelegate>
        <approvalActions>
            <action>
                <name>LeadApprovedTask1</name>
                <type>Task</type>
            </action>
            <action>
                <name>LeadApprovedTask2</name>
                <type>Task</type>
            </action>
        </approvalActions>
        <assignedApprover>
            <approver>
                <type>adhoc</type>
            </approver>
        </assignedApprover>
        <label>Step1</label>
        <name>Step1</name>
        <rejectionActions>
            <action>
                <name>LeadRejectedTask</name>
                <type>Task</type>
            </action>
        </rejectionActions>
    </approvalStep>
    <approvalStep>
        <allowDelegate>false</allowDelegate>
        <assignedApprover>
            <approver>
                <type>userHierarchyField</type>
            </approver>
        </assignedApprover>
        <entryCriteria>
            <criteriaItems>
                <field>Lead.CreatedDate</field>
                <operation>greaterThan</operation>
                <value>3/25/2013</value>
            </criteriaItems>
            <criteriaItems>
                <field>User.IsActive</field>
                <operation>notEqual</operation>
                <value>true</value>
            </criteriaItems>
        </entryCriteria>
        <ifCriteriaNotMet>ApproveRecord</ifCriteriaNotMet>
        <label>Step2</label>
        <name>Step2</name>
        <rejectBehavior>
            <type>RejectRequest</type>
```

```
        </rejectBehavior>
    </approvalStep>
    <approvalStep>
        <allowDelegate>true</allowDelegate>
        <assignedApprover>
            <approver>
                <name>MarketingTeamQueue</name>
                <type>queue</type>
            </approver>
            <approver>
                <name>LastModifiedBy</name>
                <type>relatedUserField</type>
            </approver>
            <approver>
                <name>awheeler@example.com</name>
                <type>user</type>
            </approver>
            <whenMultipleApprovers>FirstResponse</whenMultipleApprovers>
        </assignedApprover>
        <entryCriteria>
            <formula>CONTAINS( MyLeadCustomField__c , 'Salesforce')</formula>
        </entryCriteria>
        <label>Step3</label>
        <name>Step3</name>
        <rejectBehavior>
            <type>BackToPrevious</type>
        </rejectBehavior>
    </approvalStep>
    <emailTemplate>MyFolder/LeadsNewassignmentnotification</emailTemplate>
    <enableMobileDeviceAccess>false</enableMobileDeviceAccess>
    <entryCriteria>
        <criteriaItems>
            <field>Lead.AnnualRevenue</field>
            <operation>greaterThan</operation>
            <value>10500</value>
        </criteriaItems>
        <criteriaItems>
            <field>Lead.MyLeadCustomField__c</field>
            <operation>equals</operation>
            <value>Salesforce</value>
        </criteriaItems>
    </entryCriteria>
    <finalApprovalActions>
        <action>
            <name>LeadEmailContacted</name>
            <type>Alert</type>
        </action>
    </finalApprovalActions>
    <finalApprovalRecordLock>true</finalApprovalRecordLock>
    <finalRejectionActions>
        <action>
            <name>ProcessRejectedMessageAction</name>
            <type>OutboundMessage</type>
        </action>
```

```
    </finalRejectionActions>
    <finalRejectionRecordLock>false</finalRejectionRecordLock>
    <initialSubmissionActions>
        <action>
            <name>LeadFieldUpdate</name>
            <type>FieldUpdate</type>
        </action>
        <action>
            <name>NewLeadEmail</name>
            <type>Alert</type>
        </action>
    </initialSubmissionActions>
    <label>SampleProcess</label>
    <nextAutomatedApprover>
        <useApproverFieldOfRecordOwner>false</useApproverFieldOfRecordOwner>
        <userHierarchyField>customlookupuserfield__c</userHierarchyField>
    </nextAutomatedApprover>
    <postTemplate>MyPostTemplate</postTemplate>
    <recallActions>
        <action>
            <name>ProcessRecalledMessageAction</name>
            <type>OutboundMessage</type>
        </action>
    </recallActions>
    <recordEditability>AdminOnly</recordEditability>
    <showApprovalHistory>false</showApprovalHistory>
</ApprovalProcess>
```

Wildcard Support in the Manifest File

Use the wildcard character `*` (asterisk) in the `package.xml` manifest file to retrieve all approval processes for all objects. You can't use it to retrieve a subset of approval processes. Syntax such as `Lead.*` isn't supported. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

AssignmentRules

Represents assignment rules that allow you to automatically route cases to the appropriate users or queues. You can access rules metadata for all applicable objects, for a specific object, or for a specific rule on a specific object.

The `package.xml` syntax for accessing all assignment rules for all objects is:

```
    <types>
        <members>*</members>
        <name>AssignmentRules</name>
    </types>
```

All rules for a specific object use a similar syntax without the wildcard. For example, all assignment rules for the Case object would use this syntax:

```
    <types>
        <members>Case</members>
        <name>AssignmentRules</name>
    </types>
```

You can also access specific assignment rules for an object. The following example only accesses the “samplerule” and “newrule” assignment rules on the Case object. Notice that for this example the type name syntax is `AssignmentRule` and not `AssignmentRules`.

```
    <types>
        <members>Case.samplerule</members>
        <members>Case.newrule</members>
        <name>AssignmentRule</name>
    </types>
```

File Suffix and Directory Location

Assignment rules for an object have the suffix `.assignmentRules` and are stored in the `assignmentRules` folder. For example, all Case assignment rules are stored in the `Case.assignmentRules` file.

Version

AssignmentRules components are available in API version 27.0 and later.

Fields

Field Name	Field Type	Description
`assignmentRule`	AssignmentRule[]	Represents the definitions of the named assignment rules.

AssignmentRule

Specifies whether the rule is active or not and its definition. Rules are processed in the order they appear within the AssignmentRules container.

Field Name	Field Type	Description
`active`	boolean	Indicates whether the assignment rule is active (`true`) or not (`false`).
`fullname`	string	Inherited from Metadata, this field is defined in the WSDL for this metadata type. It must be specified when creating, updating, or deleting. See `createMetadata()` to see an example of this field specified for a call. This value can't be `null`.
`ruleEntry`	`RuleEntry[]`	Represents the type and description for the assignment rule.

RuleEntry

Represents the fields used by the rule.

Field Name	Field Type	Description
`assignedTo`	string	The name of the user or queue the item is assigned to.
`assignedToType`	`AssignToLookupValueType` (enumeration of type string)	Valid values are: • `User` • `Queue`
`booleanFilter`	string	Advanced filter conditions that were specified for the rule.
`criteriaItems`	`FilterItem[]`	The items in the list that define the assignment criteria.
`formula`	string	The validation formula. Specify either `formula` or `criteriaItems`, but not both fields.
`notifyCcRecipients`	boolean	Specifies whether email addresses included on the Cc line of an incoming Email-to-Case or Web-to-Lead message are included on the Cc line of the auto-response to that message (`true`) or not (`false`). Available in API version 32.0 and later.
`overrideExistingTeams`	boolean	Specifies whether the case team resets when the assignment is done `true`) or if the current team is added to the case instead of replacing the previous team (`false`).
`team`	string[]	The name of the case team. It can occur 0 or more times.
`template`	string	Specifies the template to use for the email that is automatically sent to the designated recipient. Lightning email templates aren't packageable. We recommend using a Classic email template.

Declarative Metadata Sample Definition

The following is an example file showing two assignment rules on the Case object:

```
<AssignmentRules xmlns="http://soap.sforce.com/2006/04/metadata"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <assignmentRule>
        <fullName>samplerule</fullName>
        <active>false</active>
        <ruleEntry>
            <assignedTo>testUser@org.com</assignedTo>
            <assignedToType>User</assignedToType>
            <criteriaItems>
                <field>Case.IsEscalated</field>
                <operation>equals</operation>
                <value>True</value>
            </criteriaItems>
            <template>emailtemplate</template>
```

```
        </ruleEntry>
    </assignmentRule>
    <assignmentRule>
        <fullName>Another samplerule</fullName>
        <active>false</active>
        <ruleEntry>
            <assignedTo>otherUser@org.com</assignedTo>
            <assignedToType>User</assignedToType>
            <criteriaItems>
                <field>Case.IsEscalated</field>
                <operation>equals</operation>
                <value>False</value>
            </criteriaItems>
            <template>emailtemplate</template>
        </ruleEntry>
    </assignmentRule>
</AssignmentRules>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

AssessmentQuestion

Represents the container object that stores the questions required for an assessment.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

AssessmentQuestion components have the suffix `.AssessmentQuestion` and are stored in the `AssessmentQuestions` folder.

Version

AssessmentQuestion components are available in API version 55.0 and later.

Fields

Field Name	Description
`assessmentQuestionVersion`	**Field Type** AssessmentQuestionVersion

Field Name	Description
	Description The object that stores the question versions for the assessment questions.
`dataType`	**Field Type** string **Description** Required. The data type of the assessment question.
`developerName`	**Field Type** string **Description** Required. The developer name of the assessment question. Can contain only underscores and alphanumeric characters and must be unique in your org. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
`displayTextCategory`	**Field Type** string **Description** Specifies the category of the display text when the data type is Text Block.
`formulaResponseDataType`	**Field Type** string **Description** Specifies the data type of the question response calculated by a formula.
`name`	**Field Type** string **Description** Required. The name of the record.
`questionCategory`	**Field Type** string **Description** Required. Stores the question category.

Field Name	Description
relatedQuestion	**Field Type** string **Description** Specifies the related question. Used to define a question hierarchy.

AssessmentQuestionVersion

Stores the question versions for the assessment questions.

Field Name	Description
additionalInformation	**Field Type** string **Description** The additional details for a UI element, such as the disclosure text.
description	**Field Type** string **Description** The description for the assessment question. This text isn't rendered on the assessment.
guidanceInformation	**Field Type** string **Description** The guidance for the assessment question.
helpText	**Field Type** string **Description** The text that's added as an info bubble in the UI element related to the assessment question.
isActive	**Field Type** boolean **Description** Required. Indicates whether the current version of the assessment question is set to active (`true`) or not (`false`). The default value is `false`.
name	**Field Type** string

Field Name	Description
	Description Required. Name of the assessment question version record.
`optionSourceResponseValue`	**Field Type** boolean **Description** Indicates whether the response value source for an assessment question is configured as custom (`true`) or sObject in the OmniStudio designer (`false`). The default value is `false`.
`questionText`	**Field Type** string **Description** Required. The assessment question text. Contains the label for the assessment question that appears on the assessment.
`responseValues`	**Field Type** string **Description** Holds the values to be defined in the picklist, multiselect picklist, or radio buttons.
`status`	**Field Type** string **Description** Required. Status of the assessment question version. Possible values are Draft, Active, or Archived.
`versionNumber`	**Field Type** int **Description** Required. The assessment question version number.

Declarative Metadata Sample Definition

The following is an example of an AssessmentQuestion component.

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<AssessmentQuestion
 xmlns="http://soap.sforce.com/2006/04/metadata">
 <assessmentQuestionVersion>
  <additionalInformation>ParentQuestionDevName AI</additionalInformation>
  <description>ParentQuestionDevName Desc</description>
  <helpText>ParentQuestionDevName HT</helpText>
  <isActive>true</isActive>
  <name>ParentQuestionDevName</name>
  <optionSourceResponseValue>true</optionSourceResponseValue>
  <questionText>ParentQuestionDevName Text</questionText>
  <status>Active</status>
  <versionNumber>1</versionNumber>
 </assessmentQuestionVersion>
 <dataType>DateTime</dataType>
 <developerName>ParentQuestionDevName</developerName>
 <name>ParentQuestionDevName</name>
 <questionCategory>Demographic</questionCategory>
</AssessmentQuestion>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package
 xmlns="http://soap.sforce.com/2006/04/metadata">
 <types>
  <members>*</members>
  <name>AssessmentQuestion</name>
 </types>
 <version>55.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

Usage

Before you retrieve assessment questions, we recommend that you review these considerations.

- When you retrieve an assessment question, you also get the related assessment question version with the status Active..

  Note: If an active assessment question version doesn't exist for the assessment question, then the latest assessment question version with Status as Draft is retrieved.

- The value for the `<status>` tag in the XML definition must match the status of the related assessment question version.
- If an assessment question has a related assessment question (parent question), the XML definition must include the developer name of the related assessment question.
- If the fields of an assessment question contain values, the XML definition must contain tags with those values when retrieving it.

Before you deploy assessment questions, we recommend that you review these considerations.

- If the Related Question isn't available in the target org, deploying the assessment question fails.

- If an assessment question with the same developer name exists in the target org, deploying the assessment question updates the values of the other fields in the target org.
- If the `<versionNumber>` tag is present in the XML definition of an assessment question, deploying creates a version for that question in the target org.
- If the Related Questions aren't available in target org but available in the package, then deploying the questions inserts the Related Questions in the correct order.
- If the assessment questions are associated with flows of type Discovery Framework Data Capture Flow, then deploy the assessment questions first. After deploying the assessment questions, deploy the flows.

AssessmentQuestionSet

Represents the container object for Assessment Questions.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

AssessmentQuestionSet components have the suffix `.AssessmentQuestionSet` and are stored in the `AssessmentQuestionSets` folder.

Version

AssessmentQuestionSet components are available in API version 55.0 and later.

Fields

Field Name	Description
`assessmentQuestionDeveloperNames`	**Field Type** string[] **Description** The developer names for the assessment question. Can contain only underscores and alphanumeric characters and must be unique in your org. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
`developerName`	**Field Type** string **Description** Required. The developer name for the assessment question set. Can contain only underscores and alphanumeric characters and must be unique in your org. It must begin with a

Field Name	Description
	letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
name	**Field Type** string **Description** Required. The question set name.

Declarative Metadata Sample Definition

The following is an example of an AssessmentQuestionSet component.

```
<?xml version="1.0" encoding="UTF-8"?>
<AssessmentQuestionSet
 xmlns="http://soap.sforce.com/2006/04/metadata">
 <developerName>QuestionSetDevName</developerName>
 <name>QuestionSetName</name>
 <assessmentQuestionDeveloperNames>QuestionDevName</assessmentQuestionDeveloperNames>
</AssessmentQuestionSet>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package
 xmlns="http://soap.sforce.com/2006/04/metadata">
 <types>
  <members>*</members>
  <name>AssessmentQuestion</name>
 </types>
 <types>
  <members>*</members>
  <name>AssessmentQuestionSet</name>
 </types>
 <version>55.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

Usage

Before you retrieve assessment question sets, we recommend that you review these considerations.

- When retrieving an assessment question set, if its fields contain values, then the XML definition must contain tags with those values.
- When retrieving an assessment question set, if that set is associated with multiple questions, then the XML definition must contain developer names of all the associated questions.

Before you deploy assessment question sets, we recommend that you review these considerations.

- When deploying an assessment question set, if an assessment question set with the same developer name doesn't exist in the target org, deploying creates one with that name.
- If an assessment question set with the same developer name exists in the target org, then deploying the question set updates the values of the other fields in the target org.
- If the questions associated with the assessment question set don't exist in the target org, deploying the assessment question set fails.
- If the questions associated with the assessment question set don’t exist in the target org but are available in the package, then deploying the assessment question sets inserts the questions in the correct order.

Audience

Represents the audience in an Experience Builder site. An audience consists of different types of criteria, where the audience can be assigned and used for targeting in a site. This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

Audience components have the suffix `.audience` and are stored in the `audience` folder.

Version

Audience components are available in API version 44.0 and later.

Special Access Rules

Access to the Audience type requires the AudienceMetadata permission. This permission is on by default for orgs that have Networks enabled.

Access to permission criteria for the Audience type requires the AudiencePermissionCriteria permission. This permission is available in API version 45.0 and later and is on by default for orgs that have Networks enabled.

Fields

Field Name	Field Type	Description
`audienceName`	string	Required. The name of the audience.
`container`	string	Required. The name of the site or org that contains the audience.
`criteria`	AudienceCriteria	Required. Criteria in an audience. This field is available in API version 47.0 and later.

Field Name	Field Type	Description
`criterion`	AudienceCriterion[]	Removed. List of criteria in an audience. This field is available in API version 44.0–46.0. In API version 47.0 and later, use `criteria` instead.
`description`	string	The description of the audience.
`formula`	string	Formula used to determine the audience. This field is available in API version 45.0 and later.
`formulaFilterType`	FormulaFilterType (enumeration of type string)	Indicates the audience's formula type. Valid values are • `AllCriteriaMatch` • `AnyCriterionMatches` • `CustomLogicMatches` (available in API version 45.0 and later)
`isDefaultAudience`	boolean	Indicates whether the audience is the default audience (`true`) or not (`false`). This field is available and required in API version 48.0. In API version 49.0 and later, this field is optional. The default audience file name is of format `Default_`***`Network Name.`***`audience`.
`targets`	PersonalizationTargetInfos	Targets for the audience. This field is available in API version 47.0 and later.

AudienceCriteria

Represents criteria for an audience. This subtype is available in API version 47.0 and later.

Field Name	Field Type	Description
`criterion`	AudienceCriterion[]	List of criteria for an audience. An audience can have up to 100 criteria.

AudienceCriterion

Represents a criterion for an audience.

Field Name	Field Type	Description
`criteriaNumber`	int	The number associated with the criterion in a formula, for example (1 AND 2) OR 3. This field is available in API version 45.0 and later.
`criterionValue`	AudienceCriteriaValue	The value of the criterion.
`operator`	AudienceCriterionOperator(enumeration of type string)	The operator associated with this criterion. Valid values are: • `Equal` • `NotEqual` • `GreaterThan`

Field Name	Field Type	Description
		• `GreaterThanOrEqual` • `LessThan` • `LessThanOrEqual` • `Contains` • `StartsWith` • `Includes` (available in API version 45.0 and later) • `NotIncludes` (available in API version 45.0 and later)
`type`	AudienceCriterion Type(enumeration of type string)	Required. Valid values are: • `GeoLocation` • `Domain` • `Profile` • `FieldBased` • `Permission` (available in API version 45.0 and later) • `Default` (available in API version 47.0 and later) • `Audience` (available in API version 53.0 and later) For a list of AudienceCriteriaValue fields that you can use with each AudienceCriterion `type` field value, see this table.

AudienceCriteriaValue

Represents the value of a criterion in an audience. For a list of AudienceCriteriaValue fields that you can use with each AudienceCriterion `type` field value, see this table.

Field Name	Field Type	Description
`audienceDeveloperName`	string	Developer name of the audience. This field is available in API version 53.0 and later. You can use this field only when the value of the AudienceCriterion `type` field is `Audience`.
`city`	string	City of a user. You can use this field only when the value of the AudienceCriterion `type` field is `GeoLocation`.
`country`	string	Country of a user. You can use this field only when the value of the AudienceCriterion `type` field is `GeoLocation`.
`domain`	string	Domain of a user. You can use this field only when the value of the AudienceCriterion `type` field is `Domain`.
`entityField`	string	Field of an object. You can use this field only when the value of the AudienceCriterion `type` field is `FieldBased`.
`entityType`	string	Type of object. You can use this field only when the value of the AudienceCriterion `type` field is `FieldBased`.

Field Name	Field Type	Description
`fieldValue`	string	Value of a field. You can use this field only when the value of the AudienceCriterion `type` field is `FieldBased`.
`isEnabled`	string	Indicates whether the permission is enabled (`true`) or not (`false`) for a user. This field is available in API version 45.0 and later. You can use this field used only when the value of the AudienceCriterion `type` field is `Permission`.
`permissionName`	string	Valid API name of a standard user or custom permission. This field is available in API version 45.0 and later. You can use this field only when the value of the AudienceCriterion `type` field is `Permission`.
`permissionType`	string	Type of permission. Valid values are `Standard` and `Custom`. This field is available in API version 45.0 and later. You can use this field only when the value of the AudienceCriterion `type` field is `Permission`.
`profile`	string	Profile of a user. You can use this field only when the value of the AudienceCriterion `type` field is `Profile`.
`subdivision`	string	Subdivision of a user. You can use this field only when the value of the AudienceCriterion `type` field is `GeoLocation`.

This table summarizes which AudienceCriteriaValue fields you can use with the different AudienceCriterion `type` field values.

AudienceCriterion Type	AudienceCriteriaValue Fields

`GeoLocation`	`city`
	`country`
	`subdivision`

`Domain`	`domain`
`Profile`	`profile`

`FieldBased`	`entityField`
	`entityType`
	`fieldValue`

`Permission`	`isEnabled`
	`permissionName`
	`permissionType`

`Audience`	`audienceDeveloperName`

PersonalizationTargetInfos

Represents targets for an audience. This subtype is available in API version 47.0 and later.

When deploying an audience, you must include ExperienceBundle in your package to support experience variation targets.

Field Name	Field Type	Description
target	PersonalizationTargetInfo[]	List of targets for an audience.

PersonalizationTargetInfo

Represents a target for an audience. This subtype is available in API version 47.0 and later.

Field Name	Field Type	Description
groupName	string	Required. Group name of the target. Groups bundle related target and audience pairs. You can have up to 2,000 groups and 500 targets per group. To determine the target group name, see https://developer.salesforce.com/docs/atlas.en-us.communities_dev.meta/communities_dev/communities_dev_personalization_name.htm in the *Experience Cloud Developer Guide*.
priority	int	Priority of the target. Within a group, priority determines which target is returned when the user matches more than one audience.
targetType	string	Required. Type of target, indicating the nature of the data being targeted. Supported values include: • ExperienceVariation (API version 47.0 and later) • NavigationLinkSet (API version 49.0 and later) • Report (API version 49.0 and later) • Dashboard (API version 49.0 and later) You can have up to 2,500 ExperienceVariation targets and 25,000 record targets. For more information on the ExperienceVariation target type, see ExperienceBundle.
targetValue	string	Required. Value of the target, which is the developer name of the experience variation, such as ContactSupport_ContactSupportForCalifornia_Page for a page variation. To determine the target developer name, see https://developer.salesforce.com/docs/atlas.en-us.communities_dev.meta/communities_dev/communities_dev_personalization_name.htm in the *Experience Cloud Developer Guide*.

Declarative Metadata Sample Definition

The following is an example of an Audience component.

```
<?xml version="1.0" encoding="UTF-8"?>
<Audience xmlns="http://soap.sforce.com/2006/04/metadata">
    <audienceName>Audience Metadata</audienceName>
    <container>Customer</container>
```

```
    <criteria>
        <criterion>
            <criteriaNumber>1</criteriaNumber>
            <criterionValue>
                <country>United States</country>
                <subdivision>Nevada</subdivision>
            </criterionValue>
            <operator>Equal</operator>
            <type>GeoLocation</type>
        </criterion>
        <criterion>
            <criteriaNumber>2</criteriaNumber>
            <criterionValue>
                <profile>customer community user</profile>
            </criterionValue>
            <operator>Equal</operator>
            <type>Profile</type>
        </criterion>
        <criterion>
            <criteriaNumber>3</criteriaNumber>
            <criterionValue>
                <domain>sampledomain.example.com</domain>
            </criterionValue>
            <operator>Equal</operator>
            <type>Domain</type>
        </criterion>
        <criterion>
            <criteriaNumber>4</criteriaNumber>
            <criterionValue>
              <entityField>Manager.Profile.CreatedBy.Contact.MailingCountry</entityField>

                <entityType>User</entityType>
                <fieldValue>USA</fieldValue>
            </criterionValue>
            <operator>StartsWith</operator>
            <type>FieldBased</type>
        </criterion>
        <criterion>
            <criteriaNumber>5</criteriaNumber>
            <criterionValue>
                <entityField>RecordTypeId</entityField>
                <entityType>CollaborationGroup</entityType>
                <fieldValue>CollaborationGroup.Group_RT2</fieldValue>
            </criterionValue>
            <operator>Equal</operator>
            <type>FieldBased</type>
        </criterion>
        <criterion>
            <criteriaNumber>6</criteriaNumber>
            <criterionValue>
                <isEnabled>true</isEnabled>
                <permissionName>ManageUsers</permissionName>
                <permissionType>Standard</permissionType>
            </criterionValue>
```

```
            <operator>Equal</operator>
            <type>Permission</type>
        </criterion>
        <criterion>
            <criteriaNumber>7</criteriaNumber>
            <criterionValue>
                <isEnabled>false</isEnabled>
                <permissionName>NamespaceXYZ__CustomPermABC</permissionName>
                <permissionType>Custom</permissionType>
            </criterionValue>
            <operator>Equal</operator>
            <type>Permission</type>
        </criterion>
        <criterion>
            <criteriaNumber>8</criteriaNumber>
            <criterionValue>
                <audienceDeveloperName>Audience1</audienceDeveloperName>
            </criterionValue>
            <operator>Equal</operator>
            <type>Audience</type>
        </criterion>
    </criteria>
    <formula>1 AND (2 OR 3 OR 4 OR 5 OR 6 OR 7) AND 8</formula>
    <formulaFilterType>CustomLogicMatches</formulaFilterType>
    <isDefaultAudience>false</isDefaultAudience>
    <targets>
        <target>
            <groupName>c194d79c-5c6b-4c6a-8d14-0e7042564355$#$Branding</groupName>
            <priority>1</priority>
            <targetType>ExperienceVariation</targetType>
            <targetValue>Customer_Service_testBrandingSet_Branding</targetValue>
        </target>
    </targets>
</Audience>
```

Usage

You can't use Metadata API to delete an audience.

In API version 47.0 and later, you can't create an audience without criteria.

The list of targets provided in the input for an audience is considered the state of target assignments that you want. For example, see the following information for deleting, creating, and updating targets.

If you don't have a default audience, updating targets can result in the UI erroneously showing a target assigned to the default audience. The target assignment data in the API is correct. To work around the UI issue, temporarily assign another target to the default audience and then delete it.

Personalization using audience targeting varies what the user can see in the browser but doesn't secure data in any way. To prevent users accessing sensitive data, use standard Salesforce security features, such as sharing rules and permission sets.

Delete targets

To delete a single target from an audience, deploy the entire list of targets for the audience minus the one that you want to delete.

To delete all the targets from an audience, deploy the audience with empty targets tags. For example:

```
<?xml version="1.0" encoding="UTF-8"?>
<Audience
    xmlns="http://soap.sforce.com/2006/04/metadata">
    <audienceName>testAudience</audienceName>
    <container>testContainer</container>
    <criteria>
        <criterion>
            <criteriaNumber>1</criteriaNumber>
            <criterionValue>
                <country>United States</country>
                <subdivision>Nevada</subdivision>
            </criterionValue>
            <operator>Equal</operator>
            <type>GeoLocation</type>
        </criterion>
    </criteria>
    <formulaFilterType>AllCriteriaMatch</formulaFilterType>
    <isDefaultAudience>false</isDefaultAudience>
    <targets>
    </targets>
</Audience>
```

Update an audience without updating targets

To update an audience without updating targets, deploy the audience without targets tags. For example:

```
<?xml version="1.0" encoding="UTF-8"?>
<Audience
    xmlns="http://soap.sforce.com/2006/04/metadata">
    <audienceName>testAudience</audienceName>
    <container>testContainer</container>
    <criteria>
        <criterion>
            <criteriaNumber>1</criteriaNumber>
            <criterionValue>
                <country>United States</country>
                <subdivision>Nevada</subdivision>
            </criterionValue>
            <operator>Equal</operator>
            <type>GeoLocation</type>
        </criterion>
    </criteria>
    <formulaFilterType>AllCriteriaMatch</formulaFilterType>
    <isDefaultAudience>false</isDefaultAudience>
</Audience>
```

Create targets

To create a target, deploy the entire list of targets for the audience plus the one that you want to create.

Update the priority of a target

To change the priority of a target within an audience, deploy the entire list of targets for the audience with the new priority values for the targets.

To change the priority of a target that affects priority in another audience, deploy both audiences with their entire list of targets with the new priority values for the targets.

Update the target assignment for an audience

To reassign a target to a new audience, deploy both audiences with their entire list of targets. Deploy one list with the target removed, and the other list with the target added.

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

AuraDefinitionBundle

Represents an Aura definition bundle. A bundle contains an Aura definition, such as an Aura component, and its related resources, such as a JavaScript controller. The definition can be a component, application, event, interface, or a tokens collection.

File Suffix and Directory Location

An AuraDefinitionBundle component is a folder that contains definition files. Unlike most other metadata components, an AuraDefinitionBundle component isn't a single file, it's a collection of files. Each file represents a resource in a bundle, such as markup, applications, code files (including controllers and helpers), events, documentation, and interfaces. For example, this directory structure shows the hierarchy of the folders and files for two bundles: `bundle1` and `bundle2`.

```
aura
    bundle1
        bundle1.cmp
        bundle1Controller.js
    bundle2
        bundle2.app
        bundle2Controller.js
        bundle2.auradoc
```

Aura definition bundles must be under a top-level folder named `aura`. Each bundle must have its own subfolder under the `aura` folder. The name of each definition file must start with the bundle name.

A bundle doesn't have a suffix. Definition files can have one of these suffixes:

Suffix	Component Type
`.app`	Application
`.cmp`	Component
`.design`	Design
`.evt`	Event
`.intf`	Interface
`.js`	Controller, Helper, or Renderer
`.svg`	SVG image
`.css`	Style

Suffix	Component Type
`.auradoc`	Documentation
`.tokens`	Tokens collection

Each bundle can have only one file each with a suffix of `.app`, `.cmp`, `.design`, `.evt`, `.intf`, or `.tokens`.

Version

AuraDefinitionBundle components are available in API version 32.0 and later.

Design and SVG components are available in API version 33.0 and later.

In API version 45.0 and later, there are two types of Lightning component: Aura components and Lightning web components. This metadata type describes an Aura component.

Special Access Rules

Definitions can be created only in organizations with defined namespaces.

Fields

Field Name	Field Type	Description
`apiVersion`	double	The API version for this definition bundle. When you create an Aura bundle, you can specify the API version to save it with. Available in API version 35.0 and later.
`auraDefinitions`	AuraDefinitions	Reserved for internal use.
`controllerContent`	base64Binary	The content of a JavaScript client-side controller.
`description`	string	The specification of the Aura bundle. Available in API version 35.0 and later.
`designContent`	base64Binary	The content of a design definition. Only valid inside a component bundle.
`documentationContent`	base64Binary	The content of a documentation definition.
`helperContent`	base64Binary	The content of a JavaScript helper.
`markup`	base64Binary	The content of the markup for a definition.
`modelContent`	base64Binary	Deprecated. Do not use.
`packageVersions`	PackageVersion[]	The list of installed managed package versions that this Aura definition bundle references. Available in API version 35.0 and later.
`rendererContent`	base64Binary	The content of a JavaScript client-side renderer.
`styleContent`	base64Binary	The CSS for the definition.
`SVGContent`	base64Binary	The SVG image for the definition.

Field Name	Field Type	Description
`testsuiteContent`	base64Binary	Reserved for internal use.
`type`	AuraBundleType (enumeration of type string)	The definition type. Valid values are: • `Application` • `Component` • `Event` • `Interface` • `Tokens`

Declarative Metadata Sample Definition

This example shows the directory structure of an AuraDefinitionBundle component.

```
aura
    sampleCmp
        sampleCmp.cmp
        sampleCmpController.js
```

The following samples show the contents of the metadata definition files that correspond to the sample `aura` directory.

Content of `sampleCmp.cmp`:

```
<aura:component>
    <aura:attribute name="val1" type="String" default="Value"/>
    <aura:attribute name="val2" type="String" />
    <aura:handler name="init" value="{!this}" action="{!c.myAction}"/>
        <ui:outputText value='Hello world!'/>
        <ui:outputText value='{!v.val1}'/>
        <ui:outputText value='{!v.val2}'/>
</aura:component>
```

Content of `sampleCmpController.js`:

```
({
  myAction : function(component) {
    component.set('v.val1','Value1');
    component.set('v.val2','Value2');
  }
})
```

This `package.xml` references the definitions of all Lightning components that are present in the `sampleCmp` bundle.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>sampleCmp</members>
        <name>AuraDefinitionBundle</name>
    </types>
    <version>64.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

AuthProvider

Represents an authentication provider (auth provider). An auth provider lets users log in to Salesforce from an external service provider such as Facebook, Google, or GitHub. This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

Authentication providers are stored in the `authproviders` directory. The file name matches the URL suffix, and the extension is `.authprovider`. For example, an auth provider with URL suffix `FacebookProvider` is stored in `authproviders/FacebookProvider.authprovider`.

Version

Authentication providers are available in API version 27.0 and later.

Special Access Rules

Only users with the Customize Application and Manage AuthProviders permissions can access this object.

Fields

Field Name	Field Type	Description
`appleTeam`	string	Required when using Apple as a third-party authentication provider. A 10-character team ID, obtained from an Apple developer account. Available in API version 48.0 and later.
`authorizeUrl`	string	Required when creating an OpenID Connect authentication provider. The OAuth authorization endpoint URL. Available in API version 29.0 and later. In API version 33.0 and later, for Salesforce-managed auth providers, leave the field blank to let Salesforce supply and manage the value. For details, see "Usage."
`consumerKey`	string	The app's key that is registered at the third-party (external) authentication provider. In API version 33.0 and later, for Salesforce-managed auth providers, leave the field blank to let Salesforce supply and manage the value. For details, see "Usage."
`consumerSecret`	string	The consumer secret of the app that is registered at the third-party provider. After it's set, you can't change the value. When using `create()`, this

Field Name	Field Type	Description
		field must be encrypted. To create an encrypted form of the consumer secret from plaintext: **1.** Create an authentication provider with the `consumerSecret` plaintext value. **2.** Save the authentication provider. **3.** Create an outbound change set that includes the authentication provider component. The new change set .xml file has an entry in the form `<consumerSecret>++XYZ++</consumerSecret>` where ++XYZ++ is the encrypted secret. In API version 33.0 and later, for Salesforce-managed auth providers, leave the field blank to let Salesforce supply and manage the value. For details, see Usage. If a consumer secret is defined on an authentication provider, the consumer secret is always exported as a placeholder value, not as an encrypted secret.
`controlPlane`	`MuleSoftControlPlane` (enumeration of type string)	Required when using MuleSoft as a third-party authentication provider. Environment where the MuleSoft Anypoint Platform control plane is hosted. The control plane is the part of the Anypoint Platform architecture that includes Anypoint Exchange and determines the login URL. If you select User-Specified, you must enter the Consumer Key and Consumer Secret. Obtain the values from the MuleSoft connected app that you created to store the authentication details for your Salesforce org. Available in API version 57.0 and later. Valid values include: • `None`—User-specified control plane. If you select `None`, you must enter the Consumer Key and Consumer Secret. Obtain the values from the MuleSoft connected app that you created to store the authentication details for your Salesforce org. • `US`—US control plane • `EU`—EU control plane
`customMetadataTypeRecord`	string	Required when creating a custom authentication provider plug-in. The API name of the custom authentication provider. Available in API version 36.0 and later.
`defaultScopes`	string	For OpenID Connect authentication providers, the scopes to send with the authorization request, if not specified when a flow starts. Available in API version 29.0 and later. In API version 33.0 and later, for Salesforce-managed auth providers, leave the field blank to let Salesforce supply and manage the value. See "Usage."
`ecKey`	string	Required when using Apple as a third-party authentication provider. A private key generated by Apple. Available in API version 48.0 and later.
`errorUrl`	string	A custom error URL for the authentication provider to use to report errors.

Field Name	Field Type	Description
executionUser	string	Required to specify a registration handler. The username of the Salesforce admin or system user who runs the Apex handler or flow. The execution user provides the context in which the registration handler runs. For example, if the handler creates a contact, the creation can be easily traced back to the registration process. In production, use a system user. The user must have the Manage Users permission. Available in API version 27.0 and later.
flow	string	A flow for the registration handler. The flow must be of the Identity User Registration Flow type. You can use either a flow or an Apex class for the registration handler. To use an Apex class instead, omit the `flow` field and specify an Apex class in the `registrationHandler` field. Available in API version 64.0 and later.
flowDefaultAccount	string	For authentication providers that use a flow registration handler, the default account that new external users are assigned to. If you include this field, Salesforce automatically uses it for the `defaultAccountId` variable in the Authentication Provider User Registration standard flow. A default account is required to use a flow registration handler to create and update external users. You can specify a default account here or in the flow itself. If you use both, the default account that's configured in the flow takes precedent. Available in API version 64.0 and later.
flowDefaultProfile	string	For authentication providers that use a flow registration handler, the default profile that new users are assigned to. If you include this field, Salesforce automatically uses it for the `defaultProfileId` variable in the Authentication Provider User Registration standard flow. A default profile is required to use a flow registration handler. You can specify a default profile here or in the flow itself. If you use both, the default profile that's configured in the flow takes precedent. Available in API version 64.0 and later.
friendlyName	string	Required. A user-friendly name for the authentication provider.
iconUrl	string	The path to an icon to use as a button on the login page. Users click the button to log in with the associated authentication provider, such as Twitter or Facebook. Available in API version 32.0 and later.
idTokenIssuer	string	The source of the authentication token in `https:` URI format. This field is available when configuring an OpenID Connect or Microsoft authentication provider. If provided, Salesforce validates the returned `id_token` value. OpenID Connect requires returning an `id_token` value with the `access_token` value. Available in API version 30.0 and later.

Field Name	Field Type	Description
includeOrgIdInIdentifier	boolean	Used to differentiate between users with the same user ID from two sources (such as two sandboxes). If enabled (true), Salesforce stores the org ID of the third-party identity in addition to the user ID. After you enable this setting, you can't disable it. Applies only to a Salesforce-managed auth provider. Available in API version 32.0 and later.
isPkceEnabled	boolean	Indicates whether the OAuth 2.0 Proof Key for Code Exchange (PKCE) security extension is enabled (true) or not (false). You can enable PKCE for these providerType values. • Custom • Facebook • Google • Microsoft • OpenIdConnect • Salesforce. This field is available in API version 59.0 and later.
linkKickoffUrl	string	The URL for linking existing Salesforce users to a third-party account. This field is read-only. Available in API version 43.0 and later.
logoutUrl	string	The destination for users after they log out if they authenticated using single sign-on. The URL must be fully qualified with an http or https prefix, such as https://acme.my.salesforce.com. Available in API version 33.0 and later.
oauthKickoffUrl	string	The URL for obtaining OAuth access tokens for a third party. This field is read-only. Available in API version 43.0 and later.
paramForwardAllowlist	AuthProvParamFwdAllowlist[]	An allowlisted URL parameter that can be forwarded from the authentication provider's client configuration URLs to the authorization URL. Available in API version 62.0 and later.
plugin	string	An existing Apex class that extends the Auth.AuthProviderPluginClass abstract class. Available in API version 36.0 and later.
portal	string	This field is used only with portals, which are deprecated. Salesforce doesn't support creating portals, but existing portals are supported.
providerType	AuthProviderType (enumeration of type string)	Required. The third-party authentication provider to use. Valid values include: • Apple • Bitbucket—Provides authentication for a Bitbucket provider. Enables you to connect to Bitbucket from a Lightning Platform application. When logged in to Bitbucket, the app can makes calls to Bitbucket APIs. The Bitbucket provider isn't available as an SSO provider, so users can't log in to a Salesforce org using their Bitbucket login credentials. Available in API version 61.0 and higher.

Field Name	Field Type	Description
		• `Custom`—A provider configured with a custom authentication provider plug-in. Available in API version 36.0 and later. • `Facebook`. • `GitHub`—Provides authentication for a `GitHub` provider. Used to log in users of your Lightning Platform app to GitHub using OAuth. When logged in to GitHub, your app can make calls to GitHub APIs. The `GitHub` provider isn't available as an SSO provider, so users can't log in to your Salesforce org using their GitHub login credentials. Available in API version 35.0 and later. • `Google`. • `Janrain`. • `LinkedIn`. Available in API version 32.0 and later. • `Microsoft`—Provides authentication for all services that can be accessed via Microsoft Azure Active Directory. Available in API version 55.0 and later. • `MicrosoftACS`—Microsoft Access Control Service typically provides authentication for a Microsoft Office 365 service, like SharePoint Online. The `MicrosoftACS` provider doesn't support SSO. Available in API version 31.0 and later. • `MuleSoft`. Available in API version 57.0 and later. • `OpenIdConnect`. Available in API version 29.0 and later. • `Salesforce`. • `Slack`. Available in API version 54.0 and later. • `Twitter`. Available in API version 32.0 and later.
`registrationHandler`	string	An existing Apex class that implements the `Auth.RegistrationHandler` interface. You can use either an Apex class or a flow for the registration handler. To use a flow instead, omit the `registrationHandler` field and specify a flow in the `flow` field.
`requireMfa`	boolean	Requires multi-factor authentication (MFA) for single sign-on with this auth provider based on the MFA status of each user. For this setting to trigger MFA, you must apply MFA directly to users via one of two methods. 1) Enable the org setting Require multi-factor authentication (MFA) for all direct UI logins to your Salesforce org. 2) Assign the user permission multi-factor authentication for User Interface Logins.
`sendAccessTokenInHeader`	boolean	If enabled (`true`), the access token is sent to the `UserInfoUrl` in a header instead of a query string. Available in API version 30.0 and later.
`sendClientCredentialsInHeader`	boolean	Required when creating an OpenID Connect authentication provider. If enabled (`true`), the client credentials are sent in a header to the `tokenUrl` instead of a query string. The credentials are in the standard OpenID Connect Basic Credentials header format, which is `Basic`

Field Name	Field Type	Description
		`<token>`, where `<token>` is the base64-encoded string `"clientkey:clientsecret"`. Available in API version 30.0 and later.
`sendSecretInApis`	boolean	Determines whether the encrypted consumer secret appears in API responses. If enabled (default), the secret appears in the response. If disabled (`false`), responses don't include the consumer secret. For security, you can disable the setting. However, keep in mind that: • By disabling this setting, the consumer secret is excluded from API responses in all API versions. • Change sets and other metadata deployments break because both the consumer key and secret are expected. To fix this problem, insert the consumer key manually during deployment. Available in API version 47.0 and later. The consumer secret is always included in the response as a placeholder value, regardless of the value provided for `sendSecretInApis`.
`ssoKickoffUrl`	string	The URL for performing single sign-on into Salesforce from a third party by using its third-party credentials. This field is read-only. Available in API version 43 and later.
`tokenUrl`	string	The OAuth token endpoint URL of an OpenID Connect authentication provider. Available in API version 29.0 and later. In API version 33.0 and later, for Salesforce-managed auth providers, leave the field blank to let Salesforce supply and manage the value. For details, see "Usage."
`userInfoUrl`	string	The OpenID Connect endpoint URL of the OpenID Connect authentication provider. Available in API version 29.0 and later. In API version 33.0 and later, for Salesforce-managed auth providers, leave the field blank to let Salesforce supply and manage the value. For details, see "Usage."

AuthProvParamFwdAllowlist

Represents an allowlisted URL parameter that can be forwarded from authentication provider client configuration URLs to the authorization URL. Use this type to add custom functionality to authentication providers. For example, allowlist a `ui_locales` parameter and use it to send a user's language preference from Salesforce to the third-party provider's login page.

Filed Name	Field type	Description
`description`	string	A description for the allowlisted URL parameter.

Filed Name	Field type	Description
`param`	string	The name of the parameter, such as `ui_locales` or `login_hint`.

Declarative Metadata Sample Definition

Note: Starting in November 2022, enter the `consumerSecret` value as plaintext, for example, `<consumerSecret>yourplaintextconsumersecret</consumerSecret>`. Existing consumer secrets that were entered as encrypted values can be deployed throughout the Winter '23 release.

```
<?xml version="1.0" encoding="UTF-8"?>
<AuthProvider xmlns="http://soap.sforce.com/2006/04/metadata">
    <consumerKey>yourappkey</consumerKey>
    <consumerSecret>PwdVxXjzu3NCZ3MD4He+wA==</consumerSecret>
    <executionUser>admin@your.org</executionUser>
    <friendlyName>FacebookAuthProvider</friendlyName>
    <providerType>Facebook</providerType>
    <registrationHandler>RegistrationHandler</registrationHandler>
    <sendSecretInApis>true</sendSecretInApis>
</AuthProvider>
```

This example package manifest references the previous AuthProvider definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>FacebookAuthProvider</members>
        <name>AuthProvider</name>
    </types>
    <version>28.0</version>
</Package>
```

Usage

Salesforce provides default authentication providers, called Salesforce-managed auth providers, to simplify setting up these service providers for authentication.

- Apple
- Bitbucket
- Facebook
- GitHub
- Google
- Janrain
- LinkedIn
- Microsoft
- Microsoft Access Control Service
- MuleSoft
- Salesforce

- Slack

To use a Salesforce-managed auth provider, leave these fields blank when creating your auth provider from the Auth. Provider Setup page.

- `authorizeUrl`
- `consumerKey`
- `consumerSecret`
- `defaultScopes`
- `tokenURL`
- `userInfoUrl`

Note: If you provide a value for one of these fields, you must also provide a value for `consumerKey` and `consumerSecret`.

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

AutoResponseRules

Represents an auto-response rule that sets conditions for sending automatic email responses to lead or case submissions based on the attributes of the submitted record. You can access rules metadata for all applicable objects, for a specific object, or for a specific rule on a specific object.

The `package.xml` syntax for accessing all auto-response rules for all objects is:

```
<types>
    <members>*</members>
    <name>AutoResponseRules</name>
</types>
```

All rules for a specific object use a similar syntax without the wildcard. For example, all auto-response rules for the Case object would use this syntax:

```
<types>
    <members>Case</members>
    <name>AutoResponseRules</name>
</types>
```

You can also access specific auto-response rules for an object. The following example only accesses the “samplerule” and “newrule” auto-response rules on the Case object. Notice that for this example the type name syntax is `AutoResponseRule` and not `AutoResponseRules`.

```
<types>
    <members>Case.samplerule</members>
    <members>Case.newrule</members>
    <name>AutoResponseRule</name>
</types>
```

File Suffix and Directory Location

AutoResponseRules for an object have the suffix `.autoResponseRules` and are stored in the `autoResponseRules` folder. For example, all Case auto-response rules are stored in the `Case.autoResponseRules` file.

Version

AutoResponseRules components are available in API version 27.0 and later.

Fields

Field Name	Field Type	Description
`autoresponseRule`	AutoResponseRule[]	Represents the definitions of the named auto-response rules.

AutoResponseRule

Represents whether a rule is active or not and the order in which the entry is processed in the rule.

Field Name	Field Type	Description
`active`	boolean	Indicates whether the autoresponse rule is active (`true`) or not (`false`).
`fullname`	string	Inherited from Metadata, this field is defined in the WSDL for this metadata type. It must be specified when creating, updating, or deleting. See `createMetadata()` to see an example of this field specified for a call. This value can't be `null`.
`ruleEntry`	RuleEntry[]	Represents the type and description for the auto-response rule.

RuleEntry

Represents the fields used by the rule.

Field Name	Field Type	Description
`booleanFilter`	string	Advanced filter conditions that were specified for the rule.
`criteriaItems`	`FilterItem[]`	The items in the list that define the assignment criteria.
`formula`	string	The validation formula. Specify either `formula` or `criteriaItems`, but not both fields.
`replyToEmail`	string	The email address that appears in the reply-to header.

Field Name	Field Type	Description
`senderEmail`	string	The email address of the person or queue sending the email notification.
`senderName`	string	The name of the person or queue sending the email notification.
`template`	string	Specifies the template to use for the email that is automatically sent to the designated recipient. Lightning email templates aren't packageable. We recommend using a Classic email template.

Declarative Metadata Sample Definition

The following is an example AutoResponseRules component:

```
<AutoResponseRules xmlns="http://soap.sforce.com/2006/04/metadata">
    <autoResponseRule>
        <fullName>ajbdeploytest2</fullName>
        <active>false</active>
        <ruleEntry>
            <criteriaItems>
                <field>Case.Description</field>
                <operation>contains</operation>
                <value>testing</value>
            </criteriaItems>
            <senderEmail>test@test.org</senderEmail>
            <senderName>tester name j</senderName>
            <replyToEmail>test@@test.org</replyToEmail>
            <template>emailtemplate</template>
        </ruleEntry>
    </autoResponseRule>
</AutoResponseRules>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

BatchCalcJobDefinition

Represents a Data Processing Engine definition.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

BatchCalcJobDefinition components have the suffix `.batchCalcJobDefinition` and are stored in the `batchCalcJobDefinitions` folder.

Version

BatchCalcJobDefinition components are available in API version 51.0 and later.

Special Access Rules

To use this metadata type, one of these licenses is required:

- Loyalty Management
- Financial Services Cloud
- Rebate Management
- Manufacturing Cloud
- Net Zero Cloud

Fields

Field Name	Field Type	Description
`aggregates`	BatchCalcJob Aggregate[]	Collection of aggregate nodes in a data processing engine.
`appends`	BatchCalcJobUnion[]	Collection of append nodes in a data processing engine.
`atomicWritebacks`	BatchCalcJobAtomicWriteBack[]	Collection of composite writeback nodes in a data processing engine definition. Available in API version 62.0 and later.
`customNodes`	BatchCalcJobCustomNode[]	Collection of custom nodes in a data processing engine. Available in API version 57.0 and later.
`dataSpaceApiName`	string	Stores the Data Space API Name from Data Cloud. Available in API version 60.0 and later.
`datasources`	BatchCalcJob Datasource[]	Collection of data source nodes in a data processing engine.
`definitionRunMode`	BatchCalcJobRunMode (enumeration of type string)	Specifies the execution mode in a data processing engine. Valid values are: • `Batch` • `OnDemand`
`description`	string	Description of a data processing engine definition.
`executionPlatformType`	ExecutionPlatformType (enumeration of type string)	The platform that's used to run the Data Processing Engine definition. Valid values are: • `CRMA`

Field Name	Field Type	Description
		• CDP • CORE Available in API version 59.0 and later.
filters	BatchCalcJobFilter[]	Collection of filter nodes in a data processing engine. definition.
forecasts	BatchCalcJobForecast[]	Collection of forecast nodes in a data processing engine. definition. Available in API version 58.0 and later.
hierarchyPaths	BatchCalcJobHierarchyPath[]	Collection of hierarchy path nodes in a data processing engine definition.
isTemplate	boolean	Indicates whether it's a template data processing engine definition.
joins	BatchCalcJobSourceJoin[]	Collection of join nodes in a data processing engine.
label	string	The label of a data processing engine definition.
parameters	BatchCalcJobParameter[]	Collection of input variables in a data processing engine.
processType	BatchCalcProcessType (enumeration of type string)	The process type of a data processing engine. Valid values are: • UsageManagement • AccountingPeriodClosure • AccountingSubledger—This value is reserved for internal use. • ActionableList • AdvancedAccountForecast • BenefitManagement • BillingSchedulesforInvoiceGeneration • CDPEnrichment • ChannelInventoryManagement—Available in API version 63.0 and later. • CriteriaBsdSearchAndFilter • DataProcessingEngine • DecisionMatrixDataUpload • Decisiontable • Education • EmployeeService—Available in API version 63.0 and later. • FinancialSummaryRollup • FSCHierarchyRollUp • FundraisingRollups—Available in API version 63.0 and later. • InvoiceGeneration • Loyalty

Field Name	Field Type	Description
		• `LegalEntityAccountingPeriodClosureAdvanced`—Available in API version 63.0 and later. • `LifeSciencbatchcalesCommercialTerritoryAlignment`—Available in API version 63.0 and later. • `LoyaltyPartnerManagement` • `LoyaltyPointsAggregation` • `NextGenForecasting` • `NetZero` • `PnmRosterFileUpload` • `PriceProtection` • `ProductCatalogManagement` • `ProgramBasedBusiness` • `ProviderSearch`—This value is reserved for internal use. • `Rebates` • `Recruitment` • `RevenueTransactionManagement`—Available in API version 63.0 and later. • `SalesAgreement`—Available in API version 63.0 and later. • `TestProcessType`
`status`	BatchJobDefinition Status (enumeration of type string)	Status of a data processing engine definition. Valid values are: • `Active` • `Inactive`
`transforms`	BatchCalcJobTransform[]	Collection of data transformation nodes in a data processing engine.
`writebacks`	BatchCalcJobWriteback Object[]	Collection of writeback objects in which the results of the data processing engine are written back.

BatchCalcJobAggregate

Represents a collection of fields relating to an aggregate node in a data processing engine.

Fields

Field Name	Field Type	Description
`description`	string	Description of an aggregate node.
`fields`	BatchCalcJob AggregateField[]	Required. Collection of aggregation fields.
`groupBy`	string[]	Required. Collections of fields used to group data in an aggregate node.

Field Name	Field Type	Description
label	string	Required. Label of an aggregate node.
name	string	Required. Name of an aggregate node.
sourceName	string	Required. Name of the source node.

BatchCalcJobAggregateField

Represents a collection of fields relating to an aggregation field in an aggregate node of a data processing engine.

Fields

Field Name	Field Type	Description
aggregateFunction	BatchCalcJobAggregateFunction (enumeration of type string)	Required. Function used for aggregation. Valid values are: • Unique—A count of unique values. • Sum—The sum of all values. • Max—The largest value. • Min—The smallest value. • Avg—The average value, calculated as the mean. • Std—The standard deviation. • Stdp—A standard deviation with population variance. • Var—The variance. • VarP—The variance with population. • Count—The total count of values.
alias	string	Required. Name that subsequent nodes within the data processing engine use to refer to the aggregate field.
sourceFieldName	string	Required. Source node field on which the aggregate is calculated.

BatchCalcJobAtomicWriteback

Represents a node in a DPE definition that stores the details about the relationship between the writeback nodes and the composite writeback operations between the nodes.

Field Name	Field Type	Description
description	string	Description of the composite writeback object.
label	string	Required. Name of the composite writeback object.
name	string	Required. API name of the composite writeback object.

Field Name	Field Type	Description
writebackObject Relationships	BatchCalcJobAtomicWritebackRelationship[]	Specifies the relationship between the writeback objects that are involved in the writeback operation.
writebackSequence	int	Sequence in which the data processing engine executes the composite write back node.

BatchCalcJobAtomicWritebackRelationship

Represents the relationships between the writeback objects that are involved in a composite writeback operation. It captures the relationships between these objects and the sequence in which they should be processed.

Field Name	Field Type	Description
childWriteback ObjectField	string	Field name that's associated with the child writeback object in a composite writeback relationship. Available in API version 63.0 and later.
childWriteback ObjectName	string	Name of the child writeback object that's associated with the writeback relationship.
parentWriteback ObjectField	string	Field name that's associated with the parent writeback object in a composite writeback relationship. Available in API version 63.0 and later.
parentWriteback ObjectName	string	Required. Name of the parent writeback object that's associated with the writeback relationship.
relationshipName	string	Describes the relationship between the child and parent writeback objects in a composite writeback node. Available in API version 64.0 and later.
sequenceNumber	int	Sequence number of the writeback node that's associated with its parent node in the relationship.

BatchCalcJobCustomNode

Represents a collection of custom nodes in a data processing engine. Use a custom node to add a custom action.

Fields

Field Name	Field Type	Description
description	string	Description of a custom node.
extensionName	string	Required. Name of an extension node.
extensionNamespace	string	Required. Namespace of an extension node.
label	string	Required. Label of a custom node.
name	string	Required. Name of a custom node.

Field Name	Field Type	Description
parameters	BatchCalcJob CustomNodeParameter[]	The field mappings of an extension node.
sources	string[]	Sources of an extension node.

BatchCalcJobCustomNodeParameter

Represents the field mappings of an extension node.

Fields

Field Name	Field Type	Description
name	string	Required. Name of a parameter.
value	string	Required. Value of a parameter.

BatchCalcJobDatasource

Represents a collection of fields relating to a data source node in a data processing engine.

Fields

Field Name	Field Type	Description
CSVDelimiter	BatchCalcJobCSVDelimiter (enumeration of type string)	Specifies the field separator to read fields from a CSV file record. Possible values are: • COMMA • BACKQUOTE • CARET • PIPE • SEMICOLON • TAB The default value is COMMA. The same delimiter value used for the CSV file can't be used within any of the column values in the file. If you mistakenly use the same delimiter value in column values, it can cause data parsing issues.
description	string	Description of a data source node.
fields	BatchCalcJob DatasourceField[]	Required. Collection of data source fields.

Field Name	Field Type	Description
`fileIdentifier`	string	Specifies the source of the file or file storage system.
`filePath`	string	The file path for the specified file.
`fileSource`	BatchCalcJobFileSource (enumeration of type string)	Specifies the source of the file or file storage system. Possible value is: • `ContentManagement`
`label`	string	Required. Label of a data source node.
`name`	string	Required. Name of a data source node.
`sourceName`	string	Required. Name of a standard or custom object from which the data source node extracts data.
`type`	BatchCalcJobDataSource Type (enumeration of type string)	Required. Type of object for the source object field. Supported values are: • `Analytics` • `CalculatedInsightsObject` • `CRMObject` • `CSV` • `DataModelObject` • `StandardObject` .

BatchCalcJobDatasourceField

Represents a collection of fields relating to a source object field that are selected in the data source node of a data processing engine.

Fields

Field Name	Field Type	Description
`alias`	string	Name that subsequent nodes within the data processing engine use to refer to the data source field. Required when the field name is lookup.
`dataType`	BatchCalcJobDataType (enumeration of type string)	Specifies the data type of the input field when using a CSV file as a data source. Possible values are: • `Text` • `Numeric` • `Date` • `DateTime` • `MultiValue`

Field Name	Field Type	Description
`isPrimaryKey`	boolean	Indicates whether a column name is the primary key (`true`) or not (`false`) for the Data Cloud CSV file.
`name`	string	Required. Name of the field. Can be either of the following: • Name of the source field selected in the associated data source object. • Name from a nested lookup object with three child levels.

BatchCalcJobFilter

Represents a collection of fields relating to a filter node in a data processing engine.

Fields

Field Name	Field Type	Description
`criteria`	BatchCalcJobFilterCriteria[]	Collection of filter criteria in a filter node. The field is required when `isDynamicFilter` is set to `False`.
`description`	string	Description of the batch calculation job filter.
`filterCondition`	string	Logic that is specified to apply the filter conditions. The field is required when `isDynamicFilter` is set to `False`.
`filterParameterName`	string	Name of the parameter of type filter.
`isDynamicFilter`	boolean	Indicates whether the filter criteria is dynamic. If value is set to `True`, filter criteria is passed in runtime with `filterParameterName`.
`label`	string	Required. Label of the filter node.
`name`	string	Required. Name of the filter node.
`sourceName`	string	Required. Name of the source node.

BatchCalcJobForecast

Represents a collection of fields relating to a forecast node in a data processing engine. Available in API version 58.0 and later.

Fields

Field Name	Field Type	Description
accuracyPercent	BatchCalcJobFrcstAccuracy (enumeration of type string)	The interval percentage to account for errors in forecasts. Possible values are: • `Eighty` • `NinetyFive` • `None` The default value is `None`.
aggregateFields	BtchCalcJobFrcstAggrFld[]	The list of fields to forecast.
dateFieldName	string	Required. The date field from the source node used to forecast values for the specified forecast length.
description	string	The description of the forecast node.
forecastModelType	BatchCalcJobFrcstModel (enumeration of type string)	The model used to forecast data. Possible values are: • `Additive` • `Auto` • `Multiplicative` The default value is `Auto`.
forecastPeriodCount	int	The number of time periods to generate forecast data. For example, if you select Year-Month as the forecast period type, and 4 as the forecast period count, the forecast results are generated for the next 4 months. The minimum and the default count is 1, and the maximum is 100.

Field Name	Field Type	Description
[illegible]	BatchCalcJobFrcstPeriodType (enumeration of type string)	Required. The type of forecast period to group date field values in the forecast results. Possible values are: • `FiscalYear` • `FiscalYearMonth` • `FiscalYearQuarter` • `FiscalYearWeek` • `Year` • `YearMonth` • `YearMonthDay` • `YearQuarter` • `YearWeek`
[illegible]	BatchCalcJobFrcstGrpFld[]	The source fields for grouping the data to be processed by the forecast node.
`label`	string	Required. The name of the forecast node in the UI.
`name`	string	Required. A unique name for the forecast node.
[illegible]	string	Required. The start date of the forecast period.
[illegible]	BatchCalcJobFrcstSeasonality (enumeration of type string)	Represents the periodic fluctuations that occur around the same time every year. Possible values are: • `Two` • `Three` • `Four` • `Five`

Field Name	Field Type	Description
		• `Six` • `Seven` • `Eight` • `Nine` • `Ten` • `Eleven` • `Twelve` • `Thirteen` • `Fourteen` • `Fifteen` • `Sixteen` • `Seventeen` • `Eighteen` • `Nineteen` • `Twenty` • `TwentyOne` • `TwentyTwo` • `TwentyThree` • `TwentyFour` • `Auto` • `None` The default value is `None`.
[illegible]	boolean	Indicates whether to ignore the last period in the source node when it has incomplete data (`true`) or not (`false`). The default value is `false`.
sourceName	string	Required. The name of the source node. A source can be any node other than the datasink and register node.

BtchCalcJobFrcstAggrFld

Represents a list of fields to forecast in a forecast node.

Field Name	Field Type	Description
`aggregateFunction`	BatchCalcJobAggregateFunction (enumeration of type string)	Required. The function of the aggregate field. Possible values are: • `Avg` • `Count` • `Max` • `Min` • `Std` • `StdP` • `Sum` • `Unique` • `Var` • `VarP`
`aggregationResultLabel`	string	Required. The name of the aggregation result generated from the aggregation function that's applied to the source node field.
`fieldName`	string	Required. The name of the source field.

BatchCalcJobFrcstGrpFld

Represents source fields for grouping the data to be processed by the forecast node.

Field Name	Field Type	Description
`fieldName`	string	Required. The name of the source field to group the data to be processed by the forecast node.
`groupBy`	string	A comma-separated list of values to group data by. Required when the source field type is Date or DateTime. Possible values are: • `Second` • `Second Epoch`

Field Name	Field Type	Description
		• `Minute` • `Hour` • `Day` • `Day Epoch` • `Week` • • `Month` • `Quarter` • `Year`

BatchCalcJobHierarchyPath

Represents a collection of hierarchy path nodes in a data processing engine definition.

Fields

Field Name	Field Type	Description
`description`	string	Description of the hierarchy path node.
`hierarchyFieldName`	string	Required. Field name that contains the hierarchy path.
`isSelfFieldValueIncluded`	boolean	Indicates whether the self value is included in the calculated hierarchy path (`True`) or not (`False`).
`label`	string	Required. Label of the hierarchy path node.
`name`	string	Required. Name of the hierarchy path node.
`parentFieldName`	string	Required. Parent field name to calculate hierarchy path.
`selfFieldName`	string	Required. Self field name to calculate hierarchy path.
`sourceName`	string	Required. Name of the source node.

BatchCalcJobFilterCriteria

Represents a collection of fields relating to a filter condition in a filter node in a data processing engine.

Fields

Field Name	Field Type	Description
`inputVariable`	string	Name of the input variable used as a filter.

Field Name	Field Type	Description
`operator`	BatchCalcJobFilterOperator (enumeration of type string)	Required. Operator that is specified in the filter condition. Valid values are: • `Equals` • `NotEquals` • `GreaterThan` • `GreaterThanOrEqual` • `LessThan` • `LessThanOrEqual` • `StartsWith` • `EndsWith` • `Contains` • `DoesNotContain` • `IsNull` • `IsNotNull` • `In` • `NotIn`
`sequence`	integer	Required. Sequence number used to refer the criteria in a filter node.
`sourceFieldName`	string	Required. Name of the field from the source node to apply the filter.
`value`	string	Value used to filter data from the source node.

BatchCalcJobParameter

Represents a collection of fields relating to an input variable in a data processing engine.

Fields

Field Name	Field Type	Description
`dataType`	BatchCalcJobParameterDataType (enumeration of type string)	Required. Data type of the parameter. Valid values are: • `Date` • `DateTime` • `Expression` • `FileIdentifier` • `Filter` • `Numeric` • `Text`
`defaultValue`	string	Default value of the parameter.

Field Name	Field Type	Description
`description`	string	Description of the batch calculation job parameter.
`isMultiValue`	boolean	Indicates whether the parameter has different values (`True`) or not (`False`). This field is supported only for the `Text` data type.
`label`	string	Required. Label of the batch calculation job parameter.
`name`	string	Required. Name of the batch calculation job parameter.

BatchCalcJobSourceJoin

Represents a collection of fields relating to a join node in a data processing engine.

Fields

Field Name	Field Type	Description
`description`	string	Description of the join node.
`fields`	BatchCalcJobJoinResultField[]	Collection of fields in a join node.
`joinKeys`	BatchCalcJobJoinKey[]	Collection of mapping of fields from the primary source node and the second source node in a join node.
`label`	string	Required. Label of the join node.
`name`	string	Required. Name of the join node.
`primarySourceName`	string	Required. Name associated with the node as the primary source node.
`secondarySourceName`	string	Required. Name associated with the node as the secondary source node.
`type`	BatchCalcJobSourceJoinType (enumeration of type string)	Required. Type of join specified between the primary source node and secondary source node. Valid values are: • `LeftOuter` • `RightOuter` • `Inner` • `Outer` • `Lookup`

BatchCalcJobJoinKey

Represents a collection of fields relating to a mapping of fields from the first source node and second source node in a join node of a data processing engine.

Fields

Field Name	Field Type	Description
primarySourceFieldName	string	Required. Mapped field name of the primary source node.
secondarySourceFieldName	string	Required. Mapped field name of the secondary source node.

BatchCalcJobJoinResultField

Represents a collection of fields relating to a set of resultant fields in a join node of a data processing engine.

Fields

Field Name	Field Type	Description
alias	string	Required. Name that subsequent nodes within the data processing engine definition use to refer to the resultant field.
sourceFieldName	string	Required. Name of field from the primary or secondary data source.
sourceName	string	Required. Source node of the primary or secondary data source.

BatchCalcJobTransform

Represents a collection of fields relating to a data transformation in a data processing engine.

Fields

Field Name	Field Type	Description
description	string	The description of the batch calculation job transform.
droppedFields	BatchCalcJobTransformDroppedField[]	The collection of dropped fields in a data transformation. Available when the transformation type is Slice.
expressionFields	BatchCalcJobTransformAddedField[]	The collection of formula fields in a data transformation. Available when the transformation type is Expression.
label	string	Required. The label of the batch calculation job transform.
name	string	Required. The name of the batch calculation job transform.
orderBy	BatchCalcJobOrderByField on page 395[]	A collection of fields that's used to sort the records within each partition group.
partitionBy	string[]	A group of fields that's used to partition the source data into partition groups.
sourceName	string	Required. Name of the source node.

Field Name	Field Type	Description
`transformType`	BatchCalcJobTransformType (enumeration of type string)	Required. The type of transformation. Valid values are: • `ComputeRelative`—This transformation calculates values based on values of the same partition group. • `Expression`—This transformation calculates values based on existing values of fields in the same record. • `Slice`—This transformation removes fields from the source node.

BatchCalcJobTransformDroppedField

Represents a collection of fields relating to a dropped field in a data transformation of a data processing engine.

Fields

Field Name	Field Type	Description
`sourceFieldName`	string	Required. Name of the field that is dropped.

BatchCalcJobTransformAddedField

Represents a collection of fields relating to a formula in a data transformation of a data processing engine.

Fields

Field Name	Field Type	Description
`alias`	string	Required. Name that subsequent nodes within the data processing engine use to the transform node.
`dataType`	BatchCalcJobDataType (enumeration of type string)	Required. Data type of the formula. Valid values are: • `Text` • `Numeric` • `Date` • `DateTime` • `MultiValue`
`decimalPlaces`	integer	Number of digits to the right of a decimal point in the value. Required for the `Numeric` data type.
`expression`	string	Required. Formula defined by the user.

Field Name	Field Type	Description
`length`	integer	Total length of the value including the decimal places. Required for data types: `Text` and `Numeric`.

BatchCalcJobOrderByField

Represents a collection of fields that are used to sort the partitioned data.

Fields

Field Name	Field Type	Description
`name`	string	Required. Name of the field that is used to sort data.
`orderType`	BatchCalcJobOrderType (enumeration of type string)	Order in which the data is sorted. Valid values are: • Ascending • Descending

BatchCalcJobUnion

Represents a collection of fields relating to the union of data from two nodes in a data processing engine.

Fields

Field Name	Field Type	Description
`description`	string	Description of the batch calculation job union.
`isDisjointedSchema`	boolean	Indicates whether the union is of two disjointed datasets (`true`) or not (`false`). Set to `True` to allow joining of two datasets having no common fields.
`label`	string	Required. Label of the batch calculation job union.
`name`	string	Required. Name of the batch calculation job union.
`sources`	string[]	Names of the source nodes.

BatchCalcJobWritebackObject

Represents a collection of fields relating to the object in which the results of the data processing engine are written back.

Fields

Field Name	Field Type	Description
canWrtbckToNonEditableFields	boolean	Indicates whether the non-editable fields are included in field mapping when the action type is upsert. The default value is `false`. Available in API version 64.0 and later.
description	string	Descriptions of the batch calculation job writeback object.
externalIdFieldName	string	Unique external field ID for the target object name. Available in API version 60.0 and later.
fields	BatchCalcJobWriteback Mapping[]	Collection of the writeback fields.
filterCondition	string	The condition that filters the records from a writeback dataset for a user. Examples of a filter condition include a user ID, stage name, and a security policy that returns only the records that a user owns. Available in API version 57.0 and later.
folderName	string	The folder where the writeback dataset is saved. Available in API version 57.0 and later.
isChangedRow	boolean	Indicates whether a row in the write back object is changed. Set to `True` to write back the changed rows.
isExistingDataset	boolean	Indicates whether a CRM Application (CRMA) dataset or a Data Cloud Data Lake object is present (`true`) or will be created (`false`). Available in API version 62.0 and later.
label	string	Required. Name of the write back object.
name	string	Required. Name of the batch calculation job write back object.
operationType	BatchCalcJobWriteback Opn (enumeration of type string)	Type of operation specified. Valid values are: • `Delete`—This value is available in API version 56.0 and later. • `Insert` • `Overwrite`—Available only when `storageType` is `DataLakeObject`. This value is available in API version 60.0 and later. • `Update` • `Upsert`
sharingInheritanceObjectName	string	The name of the source object from which the row-level sharing inheritance settings are applied. Available in API version 57.0 and later.
sourceName	string	Required. Name of the source node associated with the write back object.

Field Name	Field Type	Description
`storageType`	BatchCalcJobWriteback Type (enumeration of type string)	Specifies where you want to use the data stored in the source node. Available in API version 57.0 and later. Valid values are: • `Analytics` • `DataLakeObject` • `sObject` The default value is `sObject`.
`targetObjectName`	string	Required. Object that is inserted or upserted by the data processing engine.
`writebackSequence`	integer	Sequence in which the target object is updated by the data processing engine.
`writebackUser`	string	ID of the user whose permissions decide which objects and fields of the target object can be updated.

BatchCalcJobWritebackMapping

Represents a collection of fields relating to the mapping between results and the fields in the target object.

Fields

Field Name	Field Type	Description
`fieldType`	string	Target field type on the writeback object. Valid values are: • Primary Key • Qualifier Key Available in API version 64.0 and later.
isAutogenerated	boolean	Indicates whether the target field value on the writeback object is autogenerated (`true`) or not (`false`). Available in API version 64.0 and later.
`parentName`	string	Name of the lookup object. Required only when the `relationshipName` field is defined.
`relationshipName`	string	Name of the lookup relationship.
`runtimeParameter`	boolean	Indicates whether the source field from runtime parameter is `true` or `false`. The default value is `false`. Available in API version 59.0 and later.
`sourceFieldName`	string	Required. Name of the field in the source node that is written back.
`targetFieldName`	string	Name of the sObject field to which the results are written back.

Declarative Metadata Sample Definition

The following is an example of a BatchCalcJobDefinition component.

```
<?xml version="1.0" encoding="UTF-8"?>
<BatchCalcJobDefinition xmlns="http://soap.sforce.com/2006/04/metadata">
    <aggregates>
        <description>Aggregate Description</description>
        <fields>
            <aggregateFunction>Count</aggregateFunction>
            <alias>NameCount</alias>
            <sourceFieldName>Name</sourceFieldName>
        </fields>
        <groupBy>ContactId</groupBy>
        <groupBy>Name</groupBy>
        <label>AggregateOpportunities</label>
        <name>AggregateOpportunities</name>
        <sourceName>Opportunity</sourceName>
    </aggregates>
    <forecasts>
        <description>ForecastNode Description</description>
        <label>ContactForecast</label>
        <name>ContactForecast</name>
        <sourceName>Contact</sourceName>
        <dateFieldName>CreatedDate</dateFieldName>
        <forecastPeriodType>YearMonth</forecastPeriodType>
        <shouldExcludeLastPeriod>false</shouldExcludeLastPeriod>
        <forecastPeriodCount>12</forecastPeriodCount>
        <periodStartDateName>CreatedDateYM</periodStartDateName>
        <forecastModelType>Auto</forecastModelType>
        <seasonality>None</seasonality>
        <accuracyPercent>None</accuracyPercent>
        <aggregationFields>
            <aggregateFunction>Count</aggregateFunction>
            <aggregationResultLabel>CountOfLastName</aggregationResultLabel>
            <fieldName>LastName</fieldName>
        </aggregationFields>
        <groupFields>
            <fieldName>LastModifiedDate</fieldName>
            <groupBy>Week</groupBy>
        </groupFields>
    </forecasts>
    <appends>
        <description>Append desc</description>
        <isDisjointedSchema>true</isDisjointedSchema>
        <label>AppendAllAccounts</label>
        <name>AppendAllAccounts</name>
        <sources>AccountsOfManufacturingIndustry</sources>
        <sources>ComputeRelativeManufacturingIndustry</sources>
    </appends>
    <datasources>
        <description>Desc Contact</description>
        <fields>
            <alias>Id</alias>
            <name>Id</name>
```

```
            <isPrimaryKey>false</isPrimaryKey>
            <dataType>Text</dataType>
        </fields>
        <fields>
            <alias>LastName</alias>
            <name>LastName</name>
            <isPrimaryKey>false</isPrimaryKey>
            <dataType>Text</dataType>
        </fields>
        <fields>
            <alias>CreatedDate</alias>
            <name>CreatedDate</name>
            <isPrimaryKey>false</isPrimaryKey>
            <dataType>Date</dataType>
        </fields>
        <fields>
            <alias>LastModifiedDate</alias>
            <name>LastModifiedDate</name>
            <isPrimaryKey>false</isPrimaryKey>
            <dataType>Date</dataType>
        </fields>
        <label>Contact</label>
        <name>Contact</name>
        <sourceName>Contact</sourceName>
        <type>StandardObject</type>
        <fileSource>ContentManagement</fileSource>
        <fileIdentifier>069xx0000004CAeAAM</fileIdentifier>
        <CSVDelimiter>COMMA</CSVDelimiter>
        <filePath>parentFolder/childFolder</filePath>
    </datasources>
    <datasources>
        <fields>
            <alias>Name</alias>
            <name>Name</name>
            <isPrimaryKey>false</isPrimaryKey>
            <dataType>Text</dataType>
        </fields>
        <fields>
            <alias>ContactId</alias>
            <name>ContactId</name>
            <isPrimaryKey>false</isPrimaryKey>
            <dataType>Text</dataType>
        </fields>
        <label>Opportunity</label>
        <name>Opportunity</name>
        <sourceName>Opportunity</sourceName>
        <type>StandardObject</type>
        <fileSource>ContentManagement</fileSource>
        <fileIdentifier>069xx0000004CAeAAM</fileIdentifier>
        <CSVDelimiter>COMMA</CSVDelimiter>
        <filePath>parentFolder/childFolder</filePath>
    </datasources>
    <description>Calculates and creates transaction journal records based on the orders
placed by the loyalty program members. The transaction journals are used to accrue points
```

```
to the member.</description>
    <filters>
        <criteria>
            <operator>Equals</operator>
            <sequence>1</sequence>
            <sourceFieldName>LastName</sourceFieldName>
            <value>Salesforce</value>
        </criteria>
        <description>Filter Desc</description>
        <filterCondition>1</filterCondition>
        <isDynamicFilter>false</isDynamicFilter>
        <label>AccountsOfManufacturingIndustry</label>
        <name>AccountsOfManufacturingIndustry</name>
        <sourceName>AccountOpportunities</sourceName>
    </filters>
    <hierarchyPaths>
        <description>Hierarchy Path Node</description>
        <hierarchyFieldName>Hierarchy_Path</hierarchyFieldName>
        <isSelfFieldValueIncluded>true</isSelfFieldValueIncluded>
        <label>Get Hierarchy</label>
        <name>Get_Hierarchy</name>
        <parentFieldName>ContactId</parentFieldName>
        <selfFieldName>LastName</selfFieldName>
        <sourceName>AppendAllAccounts</sourceName>
    </hierarchyPaths>
    <isTemplate>false</isTemplate>
    <joins>
        <description>Left Outer Join</description>
        <fields>
            <alias>ContactId</alias>
            <sourceFieldName>Id</sourceFieldName>
            <sourceName>Contact</sourceName>
        </fields>
        <fields>
            <alias>LastName</alias>
            <sourceFieldName>LastName</sourceFieldName>
            <sourceName>Contact</sourceName>
        </fields>
        <fields>
            <alias>NameCount</alias>
            <sourceFieldName>NameCount</sourceFieldName>
            <sourceName>AggregateOpportunities</sourceName>
        </fields>
        <fields>
            <alias>OpportunityName</alias>
            <sourceFieldName>Name</sourceFieldName>
            <sourceName>AggregateOpportunities</sourceName>
        </fields>
        <joinKeys>
            <primarySourceFieldName>Id</primarySourceFieldName>
            <secondarySourceFieldName>ContactId</secondarySourceFieldName>
        </joinKeys>
        <label>AccountOpportunities</label>
        <name>AccountOpportunities</name>
```

```
        <primarySourceName>Contact</primarySourceName>
        <secondarySourceName>AggregateOpportunities</secondarySourceName>
        <type>LeftOuter</type>
    </joins>
    <label>Create Transaction Journals Based on Orders</label>
    <parameters>
        <dataType>Date</dataType>
        <defaultValue>2020-01-01</defaultValue>
        <description>Desc TextParameter</description>
        <isMultiValue>false</isMultiValue>
        <label>DateParameter</label>
        <name>DateParameter</name>
    </parameters>
    <parameters>
        <dataType>Filter</dataType>
        <defaultValue>{&quot;filterCondition&quot;: &quot;1 AND 2&quot;,
&quot;criteria&quot;: [{&quot;sourceFieldName&quot;:
&quot;NameCount&quot;,&quot;operator&quot;: &quot;GreaterThan&quot;,&quot;value&quot;:
&quot;20&quot;,&quot;sequence&quot;: &quot;1&quot;}, {&quot;sourceFieldName&quot;:
&quot;Name&quot;,&quot;operator&quot;: &quot;Equals&quot;,&quot;value&quot;:
&quot;Salesforce&quot;,&quot;sequence&quot;: &quot;2&quot;}]}</defaultValue>
        <isMultiValue>false</isMultiValue>
        <label>FilterParameter</label>
        <name>FilterParameter</name>
    </parameters>
    <parameters>
        <dataType>Numeric</dataType>
        <defaultValue>5000</defaultValue>
        <description>Desc TextParameter</description>
        <isMultiValue>false</isMultiValue>
        <label>NumericParameter</label>
        <name>NumericParameter</name>
    </parameters>
    <parameters>
        <dataType>Text</dataType>
        <defaultValue>@salesforce.com</defaultValue>
        <description>Desc TextParameter</description>
        <isMultiValue>false</isMultiValue>
        <label>TextParameter</label>
        <name>TextParameter</name>
    </parameters>
    <processType>Rebates</processType>
    <definitionRunMode>Batch</definitionRunMode>
    <status>Inactive</status>
    <transforms>
        <description>transforms Desc</description>
        <expressionFields>
            <alias>NewLastName</alias>
            <dataType>Text</dataType>
            <expression>TODAY()</expression>
            <length>80</length>
        </expressionFields>
        <label>ManufacturingIndustry</label>
        <name>ManufacturingIndustry</name>
```

```
        <sourceName>AccountsOfManufacturingIndustry</sourceName>
        <transformationType>Expression</transformationType>
    </transforms>
    <transforms>
        <droppedFields>
            <sourceFieldName>NewLastName</sourceFieldName>
        </droppedFields>
        <label>MediaIndustry</label>
        <name>MediaIndustry</name>
        <sourceName>ManufacturingIndustry</sourceName>
        <transformationType>Slice</transformationType>
    </transforms>
    <transforms>
        <description>compute relative transforms Desc</description>
        <expressionFields>
            <alias>NewLastName</alias>
            <dataType>Text</dataType>
            <expression>rank()</expression>
            <length>80</length>
        </expressionFields>
        <label>ComputeRelativeManufacturingIndustry</label>
        <name>ComputeRelativeManufacturingIndustry</name>
        <orderBy>
            <name>LastName</name>
            <orderType>Ascending</orderType>
        </orderBy>
        <partitionBy>LastName</partitionBy>
        <sourceName>MediaIndustry</sourceName>
        <transformationType>ComputeRelative</transformationType>
    </transforms>
    <customNodes>
        <name>RebatesCustomNode</name>
        <label>Rebates Custom Node</label>
        <description>customNodes Desc</description>
        <sources>Get_Hierarchy</sources>
        <extensionName>RebatesExpression</extensionName>
        <extensionNamespace>industries_mfg</extensionNamespace>
        <parameters>
            <name>inputColumn</name>
            <value>LastName</value>
        </parameters>
        <parameters>
            <name>isFilterCriteria</name>
            <value>true</value>
        </parameters>
        <parameters>
            <name>outputColumn</name>
            <value>GenName</value>
        </parameters>
    </customNodes>
    <writebacks>
        <fields>
            <sourceFieldName>GenName</sourceFieldName>
            <targetFieldName>LastName</targetFieldName>
```

```
        </fields>
        <isChangedRow>false</isChangedRow>
        <label>exportToContact</label>
        <name>exportToContact</name>
        <description>Export To Contact</description>
        <operationType>Insert</operationType>
        <sourceName>RebatesCustomNode</sourceName>
        <targetObjectName>Contact</targetObjectName>
        <writebackSequence>1</writebackSequence>
    </writebacks>
    <writebacks>
        <fields>
            <sourceFieldName>CreatedDateYM</sourceFieldName>
            <targetFieldName>CreatedDate</targetFieldName>
        </fields>
        <isChangedRow>false</isChangedRow>
        <isExistingDataset>false</isExistingDataset>
        <label>exportToContactFC</label>
        <name>exportToContactFC</name>
        <description>Export To Contact</description>
        <operationType>Insert</operationType>
        <sourceName>ContactForecast</sourceName>
        <targetObjectName>Contact</targetObjectName>
        <writebackSequence>2</writebackSequence>
    </writebacks>
</BatchCalcJobDefinition>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
   ~ Copyright 2020 Salesforce, Inc.
   ~ All Rights Reserved
   ~ Company Confidential
-->
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
   <types>
      <members>*</members>
      <name>BatchCalcJobDefinition</name>
   </types>
   <version>60.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

BatchProcessJobDefinition

Represents the details of a Batch Management job definition. This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

BatchProcessJobDefinition components have the suffix `.batchProcessJobDefinition` and are stored in the `batchProcessJobDefinitions` folder.

Version

BatchProcessJobDefinition components are available in API version 51.0 and later.

Special Access Rules

To use this metadata type, your Salesforce org must have the Loyalty Management or the Rebate Management license. The Loyalty Program Process type is only available in orgs that have Loyalty Management enabled.

Fields

Field Name	Field Type	Description
`batchSize`	integer	Required. Number of records that each Batch Management job can process. Flow type Batch Management jobs can process up to 2000 records and Loyalty Program Process type Batch Management jobs can process up to 250 records.
`dataSource`	BatchDataSource on page 405[]	Required. Source of information whose records must be processed by the Batch Management job.
`description`	string	Description of the Batch Management job, up to 255 characters.
`executionProcessApiName`	string	API name of process that must be executed by the Batch Management job. This field is available in API version 55.0 and later. • If the batch job's type is Flow, enter the API name of an active flow that the batch job must execute. • If the batch job's type is Loyalty Program Process, enter: – Transaction_Journals if you want the batch job to process Transaction Journal records by applying the applicable active loyalty program process of the type TransactionJournal. – API name of an active loyalty process of the type TierProcessing if you want the batch job to run the loyalty program process to assess the tier of eligible members. The API name consists of the name of the process, the process type, and the name of the loyalty program separated by two consecutive underscores. For example, the process API name is `Update Member Tier__TierProcessing__Inner Circle` if the

Field Name	Field Type	Description
		process name is Update Member Tier, the process type is TierProcessing, and the loyalty program name is Inner Circle. You can use database-based APEX classes that let you use flex queues in the Batch Management job, allowing to place more than 5 jobs in a queue. This functionality is applicable to all Industry Clouds that use managed packages. See Apex Flex Queue.
`flowApiName`	string	API name of an active flow process that must be executed by the Batch Management job. Note: You can either specify the flow API name in the `executionProcessApiName` field or in the `flowApiName` field.
`flowInputVariable`	string	Input variable of associated flow that is used by the batch job to uniquely identify records.
`masterLabel`	string	Required. Name of the Batch Management job, up to 80 characters.
`processGroup`	string	Required. Name of the group for which the Batch Management job processes records.
`retryCount`	integer	Required. Number of times this Batch Management job must be rerun in case it fails. The maximum retry count is 3. Valid values are 1–3.
`retryInterval`	integer	Required. Number of milliseconds after which the Batch Management job must be rerun in case it fails. Valid values are 1,000–10,000.
`status`	string	Indicates the status of the Batch Management job. Valid values are `Active` and `Inactive`.
`type`	string (enumeration of type string)	The type of process that the Batch Management job must execute. This field is available in API version 55.0 and later. Valid values are: • `Flow` • `Loyalty Program Process`

BatchDataSource

Represents the source of information whose records must be processed by the Batch Management job.

Fields

Field Name	Field Type	Description
`condition`	string	Required. Criteria defined to filter the records.
`criteria`	string	Type of filter criteria that's used to filter records for processing.

Field Name	Field Type	Description
`dataSourceType`	string	Type of data source that's used to create the batch job definition. Valid values are: • SingleSobject • MultiSobject Available in API version 64.0 and later.
`filters`	BatchDataSrcFilterCriteria on page 406[]	Filter criterion that decides which records must be processed by the Batch Management job.
orderFields	BatchDataSourceOrderField on page 407	Fields that are used to order the records before the records are added to a batch in a job.
`sourceObject`	string	Required. API name of an object whose records must be processed by the batch job. If the batch job type is Loyalty Program Process, the source object must be: • TransactionJournal if the batch job is used to process transaction journals by applying the applicable loyalty program process. • An object that stores the details of loyalty program members whose tier must be assessed by the loyalty program process specified in the executionProcessApiName field.
`sourceObjectField`	string	API name of the source object field that uniquely identifies records for which the batch job is executed. This field is available in API version 57.0 and later. This field is only applicable when the batch job's type is Loyalty Program Process and a TierProcess type active loyalty program process is specified in the `executionProcessApiName` field. Specify the API name of a field that is a lookup to the LoyaltyProgramMember object and uniquely identifies the members whose tier must be assessed.

BatchDataSrcFilterCriteria

Represents the filter conditions that decide which records must be processed by the Batch Management job.

Fields

Field Name	Field Type	Description
domainObjectName	string	Name of the object the field is associated with. Available in API version 64.0 and later.
`dynamicValueType`	string	Data type of the input variable used as a filter.
`fieldName`	string	Required. Name of the field that must be used to filter records.
`fieldPath`	string	Stores the path to a field in the object. Available in API version 64.0 and later.

Field Name	Field Type	Description
`fieldValue`	string	Required. Value of the field that must be filtered. Specify the field if `isDynamicValue` is set to `False`.
`isDynamicValue`	boolean	Required. Indicates whether the filter criteria is dynamic.
`operator`	string (enumeration of type string)	Required. Operator that is specified in the filter criteria. Valid values are: • `equals` • `excludes` • `greaterThan` • `greaterThanOrEqualTo` • `in` • `includes` • `lessThan` • `LessThanOrEqualTo` • `GreaterOrEqual` • `like` • `notEquals` • `notIn`
`sequenceNo`	integer	Required. Sequence number used to refer the criteria in a filter.

BatchDataSourceOrderField

Represents the fields that are used to group data.

Fields

Field Name	Field Type	Description
domainObjectName	string	Required. Name of the object the field is associated with. Available in API version 64.0 and later.
`fieldName`	string	Required. Name of the field that must be used to filter records. Available in API version 64.0 and later.
`fieldPath`	string	Required. Stores the path to a field in the object. Available in API version 64.0 and later.

Declarative Metadata Sample Definition

The following is an example of a BatchProcessJobDefinition component.

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<BatchProcessJobDefinition xmlns="http://soap.sforce.com/2006/04/metadata">
   <batchSize>10</batchSize>
   <dataSource>
      <condition>1</condition>
      <criteria>all</criteria>
      <filters>
         <dynamicValue>false</dynamicValue>
         <dynamicValueType>string</dynamicValueType>
         <fieldName>Name</fieldName>
         <fieldValue>abcd</fieldValue>
         <operator>equals</operator>
         <sequenceNo>1</sequenceNo>
      </filters>
      <sourceObject>Account</sourceObject>
   </dataSource>
   <flowApiName>Flow1</flowApiName>
   <flowInputVariable>recordId</flowInputVariable>
   <masterLabel>BatchJob1</masterLabel>
   <processGroup>Loyalty</processGroup>
   <retryCount>2</retryCount>
   <retryInterval>1000</retryInterval>
   <status>Inactive</status>
   <description>test</description>
   <type>Flow</type>
   <executionProcessApiName>testFlow</executionProcessApiName>
</BatchProcessJobDefinition>
```

The following is an example of a Flow object used in Metadata API.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
   ~ Copyright 2020 Salesforce, Inc.
   ~ All Rights Reserved
   ~ Company Confidential
-->
<Flow xmlns="http://soap.sforce.com/2006/04/metadata">
   <apiVersion>51.0</apiVersion>
   <interviewLabel>Flow1 {!$Flow.CurrentDateTime}</interviewLabel>
   <label>Flow1</label>
   <processMetadataValues>
      <name>BuilderType</name>
      <value>
         <stringValue>LightningFlowBuilder</stringValue>
      </value>
   </processMetadataValues>
   <processMetadataValues>
      <name>OriginBuilderType</name>
      <value>
         <stringValue>LightningFlowBuilder</stringValue>
      </value>
   </processMetadataValues>
   <processType>AutoLaunchedFlow</processType>
   <recordLookups>
      <name>getAcc</name>
```

```
      <label>getAcc</label>
      <locationX>614</locationX>
      <locationY>465</locationY>
      <assignNullValuesIfNoRecordsFound>false</assignNullValuesIfNoRecordsFound>
      <filterLogic>and</filterLogic>
      <filters>
         <field>Id</field>
         <operator>EqualTo</operator>
         <value>
            <elementReference>recordId</elementReference>
         </value>
      </filters>
      <getFirstRecordOnly>true</getFirstRecordOnly>
      <object>Account</object>
      <storeOutputAutomatically>true</storeOutputAutomatically>
   </recordLookups>
   <start>
      <locationX>73</locationX>
      <locationY>213</locationY>
      <connector>
         <targetReference>getAcc</targetReference>
      </connector>
   </start>
   <status>Draft</status>
   <variables>
      <name>recordId</name>
      <dataType>String</dataType>
      <isCollection>false</isCollection>
      <isInput>true</isInput>
      <isOutput>false</isOutput>
   </variables>
</Flow>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
   <types>
      <members>*</members>
      <name>BatchProcessJobDefinition</name>
   </types>
   <types>
      <members>Flow1</members>
      <name>Flow</name>
   </types>
   <version>51.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

BillingSettings

Represents the settings for Salesforce Billing.

Parent Type and Manifest Access

This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all the settings metadata types for the org are accessed using the "Settings" name. See Settings for more details.

File Suffix and Directory Location

The `BillingSettings` values are stored in the `BillingSettings.settings` file in the `settings` folder. The `.settings` files are different from other named components, because there's only one settings file for each settings component.

Version

BillingSettings components are available in API version 62.0 and later.

Special Access Rules

These settings are available when Billing is enabled.

Fields

Field Name	Description
`enableBillingSetup`	**Field Type** boolean **Description** Indicates whether to enable Billing setting (`true`) or not (`false`). The default value is `false`.
`enableCrMemoApplicationToPostedInvoices`	**Field Type** boolean **Description** Indicates whether to enable Apply Credits to Posted Invoices setting (`true`) or not (`false`). The default value is `false`. This setting automates settlement of invoices through application of credits to posted invoices. The credit application level determines whether credits are automatically applied to invoices or invoice lines.
`enableInvoiceEmailDelivery`	**Field Type** boolean

Field Name	Description
	Description Indicates whether to enable Configure Email Delivery Settings (`true`) or not (`false`). The default value is `false`. Available in API version 63.0 and later with Revenue Cloud Billing.
`enableInvoicePdfGeneration`	**Field Type** boolean **Description** Indicates whether to enable Document Generation setting (`true`) or not (`false`). The default value is `false`. Available in API version 63.0 and later with Revenue Cloud Billing.
`enableNegInvoiceLn ConversionToCrMemoLn`	**Field Type** boolean **Description** Indicates whether to enable Convert Negative Invoice Lines to Credit Memo Lines setting (`true`) or not (`false`). The default value is `false`.
`enablePaymentSchedulesAnd ItemsCreation`	**Field Type** boolean **Description** Indicates whether to create a default payment schedule policy and payment schedule treatment (`true`) or not (`false`). The default value is `false`. If enabled, payment schedules and payment schedule items are created during financial transactions such as posting of invoices. Available in API version 64.0 and later with Revenue Cloud Billing.
`enableTransactions ApplicationToInvoices`	**Field Type** boolean **Description** Indicates whether to enable Credit Application Level setting (`true`) or not (`false`). The default value is `false`. **Revenue Cloud Advanced** This setting applies balances of credit memos to invoices or balances of credit memo lines to invoice lines. For the latter, amounts and balances on the invoices are rolled-up from the related invoice lines. **Revenue Cloud Billing** This setting applies balances of credit memos and payments to invoices or balances of credit memo lines and payments lines to invoice lines. For the latter, amounts and balances on the invoices are rolled-up from the related invoice lines.

Field Name	Description
`enableTransactionJournal Creation`	**Field Type** boolean **Description** Indicates whether to create Transaction Journal records based on the defined general ledger account assignment rules for the billing entities when billing transaction records are created or updated (`true`) or not (`false`). The default value is `false`. Available in API version 63.0 and later with Revenue Cloud Billing. Billing transaction records include these transaction types. • Invoice • Invoice Line • Invoice Line Tax • Credit Memo • Credit Memo Line • Credit Memo Line Tax • Payment • Refund • Payment Line Invoice • Payment Line Invoice Line • Credit Memo Inv Application • Credit Memo Line Invoice Line
`enableTrxnAmountsStorage InCorpCurrency`	**Field Type** boolean **Description** Indicates whether to allow conversion of amounts of the Invoice, Invoice Line, Credit Memo, and Credit Memo Line records to your corporate currency (`true`) or not (`false`). The default value is `false`. Available in API version 63.0 and later. Store the converted amounts in corporate currency-specific amount fields.

Declarative Metadata Sample Definition

The following is an example of a BillingSettings component.

```
<BillingSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <enableBillingSetup>true</enableBillingSetup>
    <enableInvoicePdfGeneration>true</enableInvoicePdfGeneration>
   <enableTransactionsApplicationToInvoices>true</enableTransactionsApplicationToInvoices>

   <enableCrMemoApplicationToPostedInvoices>true</enableCrMemoApplicationToPostedInvoices>

    <enableInvoiceEmailDelivery>true</enableInvoiceEmailDelivery>
    <enableTransactionJournalCreation>true</enableTransactionJournalCreation>
```

```
    <enableTrxnAmountsStorageInCorpCurrency>true</enableTrxnAmountsStorageInCorpCurrency>

    <enablePaymentSchedulesAndItemsCreation>true</enablePaymentSchedulesAndItemsCreation>
</BillingSettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>Billing</members>
        <name>Settings</name>
    </types>
    <version>64.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

BlacklistedConsumer

Represents a connected app that is inaccessible to your Salesforce org's users. This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

BlacklistedConsumer components have the suffix `.blacklistedConsumer` and are stored in the `blacklistedConsumers` folder.

Version

BlacklistedConsumer components are available in API version 49.0 and later.

Fields

This metadata type contains the following fields:

Field Name	Field Type	Description
`blockedByApiWhitelisting`	boolean	Set to `true` to apply the Permitted Users policy, `Admin approved users are pre-authorized` to all connected apps in the org. This policy limits access to only users with the associated profile or

Field Name	Field Type	Description
		permission set assigned to the app. Set to `false` to allow access to the connected app. False is the default value.
`consumerKey`	string	Required. A value used by the consumer for identification of the connected app to Salesforce. Referred to as `client_id` in OAuth 2.0. After you define and save the value, it can't be edited. The value must be alphanumeric, can't contain special characters or spaces, and must be between 8–256 characters. Consumer keys must be globally unique.
`consumerName`	string	Required. The name of the connected app being blocked.
`masterLabel`	string	Required. The primary label for the connected app record.

Declarative Metadata Sample Definition

The following is an example of a component.

```
<BlacklistedConsumer xmlns="http://soap.sforce.com/2006/04/metadata">
    <consumerKey>testConsumerKey</consumerKey>
    <consumerName>testName</consumerName>
    <blockedByApiWhitelisting>false</blockedByApiWhitelisting>
    <masterLabel>myTest</masterLabel>
</BlacklistedConsumer>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>BlacklistedConsumer</name>
    </types>
    <version>49.0</version>
</Package>
```

Usage

Use this type judiciously for connected apps that you want to make inaccessible to your org's users. Blocking an app ends all current user sessions and prevents future sessions. To block malicious attempts to access your org's data, we recommend using API Access Control instead. This feature restricts users from accessing your Salesforce APIs unless they are pre-authorized through an approved connected app.

Bot

Represents a definition of an Einstein Bot configuration that can have one or more versions. Only one version can be active.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

Bot components have the suffix `.bot` and are stored in the `bots` folder.

Version

Bot components are available in API version 43.0 and later.

Special Access Rules

Bot is available only if Chat and Einstein Bots are enabled in your org.

Fields

Field Name	Field Type	Description
`botMlDomain`	LocalMlDomain on page 416	Represents the Einstein intent set that groups intents, entities, and variables associated with a bot. All Einstein Bot versions under the same bot now share an intent set. Available in API version 44.0 and later.
`botUser`	string	Represents a user profile associated with the bot. Available in API version 46.0 and later.
`botVersions`	BotVersion on page 435	Represents the configuration details for a specific Einstein Bots version, including dialogs, intents, entities, and variables.
`contextVariables`	ConversationContextVariable on page 416	Represents the context variables that enable your bot to gather customer information regardless of channel. Available in API 45.0 and later.
`conversationChannelProviders`	ConversationDefinitionChannelProvider[] on page 418	Represents a list of the conversation channels linked to the bot. Available in API version 51.0 and later.
`description`	string	A description of the bot.
`label`	string	Label that identifies the bot throughout the Salesforce user interface.
`logPrivateConversationData`	boolean	Specifies whether to log customer inputs as part of conversation data (`true`) or not (`false`). Available in API version 48.0 and later.
sessionTimeout	int	Represents the maximum amount of minutes that a bot session can be idle. Available in API version 58.0 and later.
`type`	BotType (enumeration of type string)	Required. The default value is `Bot`. This field represents the configuration type of the bot. Valid values are: • `Bot`— Default Einstein Bot configuration. • `ExternalCopilot`— An external-facing agent. For example, an Agentforce Service agent.

Field Name	Field Type	Description
		• `InternalCopilot`— An internal-facing agent. For example, an Agentforce Employee agent.

LocalMlDomain

An Einstein Intent Set local to the current bot version.

Field Name	Field Type	Description
`label`	string	Label that represents an Einstein Intent Set local to the current bot version throughout the Salesforce user interface.
`mlIntents`	MlIntent[]	List of intents associated with this local intent set.
`mlSlotClasses`	MlSlotClass[]	List of entities associated with this local intent set.
`name`	string	Required. This unique name prevents conflicts with other local Einstein Intent Sets. This name can contain only underscores and alphanumeric characters and must be unique in your org. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.

ConversationContextVariable

A context variable local to the current bot version. Available in API version 45.0 and later.

Field Name	Field Type	Description
`contextVariableMappings`	ConversationContextVariableMapping on page 417	Represents the mapping between a context variable, channel type, and sObject field.
`dataType`	ConversationDataType (enumeration of type string)	Required. Represents the data type of the context variable. Valid values are: • `Text` • `Number` • `Boolean` • `Object` • `Date` • `DateTime` • `Currency` • `Id`
`description`	string	A description of this variable. This value may be used by the Agentforce planner service. Available in API version 63.0 and later.
`developerName`	string	Required. Represents the name of the context variable. Can contain only underscores and alphanumeric characters and must be unique in your org. It

Field Name	Field Type	Description
		must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
`includeInPrompt`	boolean	Indicates whether the variable is injected into the prompt sent to the Agentforce model. Available in API version 63.0 and later.
`label`	string	Required. A label that identifies the context variable throughout the Salesforce user interface.
`SObjectType`	string	Valid values are: • `BotDefinition` • `Queue`

ConversationContextVariableMapping

Represents the mapping between a context variable, channel type, and sObject field.

Field Name	Field Type	Description
`fieldName`	string	Required. The API name of an SObject field to be used as part of the mapping.
`messageType`	MessageType (enumeration of type string)	Required. Represents the message channel. Valid values are: • `Alexa` • `AppleBusinessChat`—Messages sent in enhanced Apple Messages for Business channels. • `EmbeddedMessaging`—Messages sent in Messaging for In-App and Web channels. Available in API version 50.0 and later. • `Facebook` • `GoogleHome` • `InternalCopilot` • `Line` • `Omega` • `Phone` • `Text` • `WeChat` • `WebChat` • `WhatsApp`
`SObjectType`	string	Required. SObject type for the field property defined as part of the mapping. Valid values are: • `LiveChatTranscript` • `MessagingEndUser` • `MessagingSession`

ConversationDefinitionChannelProvider

The developer name of a conversation channel linked to the bot. Available in API version 51.0 and later.

Note: To add, edit, or remove a messaging channel, you must use the UI. If you deploy a bot with messaging channel providers, those providers aren't visible in Metadata API.

Field Name	Field Type	Description
`agentRequired`	boolean	Specifies whether an agent must be online for the bot to be active (`true`) or not (`false`) The default is `false`.
`chatButtonName`	string	Required. The developer name of a LiveChatButton metadata component.

Declarative Metadata Sample Definition

The following is an example of a Bot. This example has been trimmed to make it easier to read.

```
<?xml version="1.0" encoding="UTF-8"?>
<Bot xmlns="http://soap.sforce.com/2006/04/metadata">
    <botMlDomain>
        <label>Astros Pizza</label>
        <mlIntents>
            <developerName>New_Order</developerName>
            <label>New Order</label>
            <mlIntentUtterances>
                <utterance>Today is pie day so I want pie</utterance>
            </mlIntentUtterances>
        </mlIntents>
        <mlSlotClasses>
            <developerName>Size</developerName>
            <extractionType>Value</extractionType>
            <label>Size</label>
            <mlSlotClassValues>
                <synonymGroup>
                    <languages>en_US</languages>
                    <terms>Big</terms>
                    <terms>Extra Large</terms>
                    <terms>X-Large</terms>
                    <terms>Grande</terms>
                    <terms>Huge</terms>
                </synonymGroup>
                <value>Large</value>
            </mlSlotClassValues>
        </mlSlotClasses>
        <name>Astros_Pizza_ld1</name>
    </botMlDomain>
    <botVersions>
        <fullName>v1</fullName>
        <botDialogGroups>
            <developerName>Order_Management</developerName>
            <label>Order Management</label>
        </botDialogGroups>
```

```
        <botDialogs>
            <botDialogGroup>Order_Management</botDialogGroup>
            <botSteps>
                <botMessages>
                    <message>□□□□□□Pizza Time! □□□□□□</message>
                </botMessages>
                <type>Message</type>
            </botSteps>
            <botSteps>
                <botStepConditions>
                    <leftOperandName>Verified_User</leftOperandName>
                    <leftOperandType>ConversationVariable</leftOperandType>
                    <operatorType>Equals</operatorType>
                    <rightOperandValue>false</rightOperandValue>
                </botStepConditions>
                <botSteps>
                    <botNavigation>
                        <botNavigationLinks>
                            <targetBotDialog>Customer_Verification</targetBotDialog>
                        </botNavigationLinks>
                        <type>Call</type>
                    </botNavigation>
                    <type>Navigation</type>
                </botSteps>
                <type>Group</type>
            </botSteps>
            <botSteps>
                <botStepConditions>
                    <leftOperandName>Location</leftOperandName>
                    <leftOperandType>ConversationVariable</leftOperandType>
                    <operatorType>IsNotSet</operatorType>
                </botStepConditions>
                <botSteps>
                    <botNavigation>
                        <botNavigationLinks>
                            <targetBotDialog>Select_Location</targetBotDialog>
                        </botNavigationLinks>
                        <type>Call</type>
                    </botNavigation>
                    <type>Navigation</type>
                </botSteps>
                <type>Group</type>
            </botSteps>
            <botSteps>
                <botVariableOperation>
                    <botInvocation>
                        <invocationActionName>CreateOrderService</invocationActionName>
                        <invocationActionType>apex</invocationActionType>
                        <invocationMappings>
                            <parameterName>customer</parameterName>
                            <type>Input</type>
                            <variableName>Contact</variableName>
                            <variableType>ConversationVariable</variableType>
                        </invocationMappings>
```

```
                        <invocationMappings>
                            <parameterName>location</parameterName>
                            <type>Input</type>
                            <variableName>Location</variableName>
                            <variableType>ConversationVariable</variableType>
                        </invocationMappings>
                        <invocationMappings>
                            <parameterName>output</parameterName>
                            <type>Output</type>
                            <variableName>Pizza_Order</variableName>
                            <variableType>ConversationVariable</variableType>
                        </invocationMappings>
                    </botInvocation>
                    <type>Set</type>
                </botVariableOperation>
                <type>VariableOperation</type>
            </botSteps>
            <botSteps>
                <botMessages>
                 <message>Perfect, let&apos;s work on your order from our {!Location.Name}
location</message>
                </botMessages>
                <type>Message</type>
            </botSteps>
            <botSteps>
                <botNavigation>
                    <botNavigationLinks>
                        <targetBotDialog>Add_Items_to_Order</targetBotDialog>
                    </botNavigationLinks>
                    <type>Redirect</type>
                </botNavigation>
                <type>Navigation</type>
            </botSteps>
            <developerName>New_Order</developerName>
            <label>New Order</label>
            <mlIntent>New_Order</mlIntent>
            <showInFooterMenu>false</showInFooterMenu>
        </botDialogs>
        <conversationVariables>
            <dataType>Object</dataType>
            <developerName>Contact</developerName>
            <label>Contact</label>
        </conversationVariables>
        <conversationVariables>
            <dataType>Text</dataType>
            <developerName>Delivery_Address</developerName>
            <label>Delivery Address</label>
        </conversationVariables>
        <conversationVariables>
            <dataType>Object</dataType>
            <developerName>Pizza_Order</developerName>
            <label>Pizza Order</label>
        </conversationVariables>
        <entryDialog>Welcome</entryDialog>
```

```
        <mainMenuDialog>Main_Menu</mainMenuDialog>
    </botVersions>
    <contextVariables>
        <contextVariableMappings>
            <SObjectType>LiveChatTranscript</SObjectType>
            <fieldName>LiveChatTranscript.ChatKey</fieldName>
            <messageType>WebChat</messageType>
        </contextVariableMappings>
        <dataType>Text</dataType>
        <developerName>ChatKey</developerName>
        <label>Chat Key</label>
    </contextVariables>
    <contextVariables>
        <contextVariableMappings>
            <SObjectType>LiveChatTranscript</SObjectType>
            <fieldName>LiveChatTranscript.ContactId</fieldName>
            <messageType>WebChat</messageType>
        </contextVariableMappings>
        <dataType>Id</dataType>
        <developerName>ContactId</developerName>
        <label>Contact Id</label>
    </contextVariables>
    <contextVariables>
        <contextVariableMappings>
            <SObjectType>LiveChatTranscript</SObjectType>
            <fieldName>LiveChatTranscript.LiveChatVisitorId</fieldName>
            <messageType>WebChat</messageType>
        </contextVariableMappings>
        <dataType>Id</dataType>
        <developerName>EndUserId</developerName>
        <label>End User Id</label>
    </contextVariables>
    <contextVariables>
        <contextVariableMappings>
            <SObjectType>LiveChatTranscript</SObjectType>
            <fieldName>LiveChatTranscript.Id</fieldName>
            <messageType>WebChat</messageType>
        </contextVariableMappings>
        <dataType>Id</dataType>
        <developerName>RoutableId</developerName>
        <label>Routable Id</label>
    </contextVariables>
....<conversationChannelProviders>
        <agentRequired>false</agentRequired>
        <chatButtonName>Chat_Button_For_Bot</chatButtonName>
    </conversationChannelProviders>
    <label>Astro&amp;apos;s Pizza</label>
</Bot>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>Pizza_Bot</members>
```

```
        <name>Bot</name>
    </types>
    <version>45.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

BotBlock

Represents the configuration details for a specific Einstein Bot block, including dialogs and variables.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

BotBlock components have the suffix `.botBlock` and are stored in the `botBlocks` folder.

Version

BotBlock components are available in API version 58.0 and later.

Special Access Rules

BotBlock is available only if Chat and Einstein Bots are enabled in your org.

Fields

Field Name	Description
`botBlockVersions`	**Field Type** BotBlockVersion[] **Description** The configuration details for specific Einstein Bot block versions, including dialogs and variables.
`description`	**Field Type** string

Field Name	Description
	Description A description of the bot block.
`masterLabel`	**Field Type** string **Description** Required. A user-friendly label for BotBlock, which is defined when the block is created.
`richContentEnabled`	**Field Type** boolean **Description** Indicates whether the block is available for enhanced bots (`true`) or for only standard bots (`false`). The default is `false`.

BotBlockVersion

Represents the configuration details for an Einstein Bot block version, including dialogs and variables.

Field Name	Description
`botDialogs`	**Field Type** BotDialog[] on page 439 **Description** The list of dialogs in this bot block.
`conversationGoals`	**Field Type** ConversationDefinitionGoal[] on page 458 **Description** The list of goals in this bot block. Available in API version 57.0 and later.
`conversationLanguages`	**Field Type** string **Description** Required. Specifies the language of the bot block.
`conversationVariables`	**Field Type** ConversationVariable[] on page 460

Field Name	Description
	Description A container that stores a specific piece of data collected from the customer. You can use variables within dialog actions as both inputs and outputs. Available in API version 44.0 and later.
`description`	**Field Type** string **Description** A description of the bot block.
`mlDomain`	**Field Type** LocalMlDomain on page 416 **Description** Required. The Einstein Intent Set that groups intents, entities, and variables associated with a block.
`permissionSet`	**Field Type** string **Description** The permission set associated with the bot block. Available in API version 59.0 and later.
`status`	**Field Type** ConvDefBlockVersionStatus (enumeration of type string) **Description** Required. Indicates whether a block is published or is a draft. Values are: • `Published`

Declarative Metadata Sample Definition

The following is an example of a BotBlock component.

```
<?xml version="1.0" encoding="UTF-8"?>
<BotBlock xmlns="http://soap.sforce.com/2006/04/metadata">
    <access xsi:nil="true"/>
    <botBlockVersions>
        <fullName>Published</fullName>
        <botDialogs>
            <developerName>Test_Dialog_1646070168572</developerName>
```

```
            <label>Test_Dialog_1646070168572</label>
            <showInFooterMenu>false</showInFooterMenu>
        </botDialogs>
        <botDialogs>
            <developerName>Test_Dialog_1646070168926</developerName>
            <label>Test_Dialog_1646070168926</label>
            <showInFooterMenu>false</showInFooterMenu>
        </botDialogs>
        <botDialogs>
            <botSteps>
                <stepIdentifier>s4</stepIdentifier>
                <type>Wait</type>
            </botSteps>
            <developerName>Main_Menu</developerName>
            <label>Main Menu</label>
            <mlIntent>Main_Menu</mlIntent>
            <showInFooterMenu>false</showInFooterMenu>
        </botDialogs>
        <botDialogs>
            <botSteps>
                <botMessages>
                    <message>Goodbye! Click the &amp;quot;End Chat&amp;quot; button to end this
chat</message>
                    <messageIdentifier>m2</messageIdentifier>
                </botMessages>
                <stepIdentifier>s6</stepIdentifier>
                <type>Message</type>
            </botSteps>
            <botSteps>
                <stepIdentifier>s7</stepIdentifier>
                <type>Wait</type>
            </botSteps>
            <developerName>End_Chat</developerName>
            <label>End Chat</label>
            <mlIntent>End_Chat</mlIntent>
            <showInFooterMenu>false</showInFooterMenu>
        </botDialogs>
        <botDialogs>
            <botSteps>
                <botMessages>
                    <message>Unfortunately, there are no agents available at the
moment</message>
                    <messageIdentifier>m3</messageIdentifier>
                </botMessages>
                <stepIdentifier>s8</stepIdentifier>
                <type>Message</type>
            </botSteps>
            <botSteps>
                <stepIdentifier>s9</stepIdentifier>
                <type>Wait</type>
            </botSteps>
            <developerName>No_Agent_Available</developerName>
            <label>No Agent</label>
            <showInFooterMenu>false</showInFooterMenu>
```

```
    </botDialogs>
    <botDialogs>
        <botSteps>
            <botMessages>
                <message>Hi! I&amp;apos;m your helpful bot.</message>
                <messageIdentifier>m1</messageIdentifier>
            </botMessages>
            <stepIdentifier>s1</stepIdentifier>
            <type>Message</type>
        </botSteps>
        <botSteps>
            <conversationRecordLookup>
                <SObjectType>Account</SObjectType>
                <conditions>
                    <leftOperand>Account.Phone</leftOperand>
                    <operatorType>Equal</operatorType>
                    <rightOperandValue>Value</rightOperandValue>
                    <sortOrder>0</sortOrder>
                </conditions>
                <lookupFields>
                    <fieldName>Account.Phone</fieldName>
                </lookupFields>
                <lookupFields>
                    <fieldName>Account.OwnerId</fieldName>
                </lookupFields>
                <maxLookupResults>1</maxLookupResults>
                <sourceVariableName>_LastCustomerInput</sourceVariableName>
                <sourceVariableType>ConversationVariable</sourceVariableType>
                <targetVariableName>MyCustomVariable</targetVariableName>
            </conversationRecordLookup>
            <stepIdentifier>s2</stepIdentifier>
            <type>RecordLookup</type>
        </botSteps>
        <botSteps>
            <botNavigation>
                <botNavigationLinks>
                    <targetBotDialog>Main_Menu</targetBotDialog>
                    <targetVariable xsi:nil="true"/>
                    <targetVariableType xsi:nil="true"/>
                </botNavigationLinks>
                <type>Redirect</type>
            </botNavigation>
            <stepIdentifier>s3</stepIdentifier>
            <type>Navigation</type>
        </botSteps>
        <developerName>Welcome</developerName>
        <label>Welcome</label>
        <mlIntent>Welcome</mlIntent>
        <showInFooterMenu>false</showInFooterMenu>
    </botDialogs>
    <conversationLanguages>en_US</conversationLanguages>
    <conversationVariables>
        <dataType>Text</dataType>
        <developerName>TestVariableABC</developerName>
```

```
            <label>TestVariableABC</label>
        </conversationVariables>
        <conversationVariables>
            <dataType>Text</dataType>
            <developerName>TestVariableXYZ</developerName>
            <label>TestVariableXYZ</label>
        </conversationVariables>
        <conversationVariables>
            <collectionType>List</collectionType>
            <dataType>Object</dataType>
            <developerName>MyCustomVariable</developerName>
            <label>MyCustomVariable</label>
        </conversationVariables>
        <description>Created for testing.</description>
        <mlDomain>
            <label>vPub</label>
            <mlIntents>
                <developerName>End_Chat</developerName>
                <label>End Chat</label>
                <mlIntentUtterances>
                    <language>es</language>
                    <utterance>Utterance1</utterance>
                </mlIntentUtterances>
                <mlIntentUtterances>
                    <language>es</language>
                    <utterance>Utterance2</utterance>
                </mlIntentUtterances>
                <mlIntentUtterances>
                    <language>es</language>
                    <utterance>Utterance3</utterance>
                </mlIntentUtterances>
            </mlIntents>
            <mlIntents>
                <description>Main Menu Intent</description>
                <developerName>Main_Menu</developerName>
                <label>Main Menu</label>
            </mlIntents>
            <mlIntents>
                <description>Welcome Intent</description>
                <developerName>Welcome</developerName>
                <label>Welcome</label>
            </mlIntents>
            <name>blockDevName0001_vPub</name>
        </mlDomain>
        <status>Published</status>
    </botBlockVersions>
    <description>Collects the user&amp;apos;s first name, last name, email address, phone
number, and company name.</description>
    <masterLabel>User Info Collection Block</masterLabel>
    <richContentEnabled>true</richContentEnabled>
</BotBlock>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>AgentTransfer</members>
        <name>BotBlock</name>
    </types>
    <version>58.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

BotTemplate

Represents the configuration details for a specific Einstein Bot template, including dialogs and variables.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

BotTemplate components have the suffix `.botTemplate` and are stored in the `botTemplates` folder.

Version

BotTemplate components are available in API version 55.0 and later.

Special Access Rules

BotTemplate is available only if Chat and Einstein Bots are enabled in your org.

Fields

Field Name	Description
`botDialogGroups`	**Field Type** BotDialogGroup[] on page 438 **Description** The list of dialog groups in this bot template.

Field Name	Description
botDialogs	**Field Type** BotDialog[] on page 439 **Description** The list of dialogs in this bot template.
conversationLanguages	**Field Type** string **Description** Required. Specifies the language of the bot template.
contextVariables	**Field Type** ConversationContextVariable[] on page 416 **Description** Represents the context variables that enable your bot to gather customer information regardless of channel.
conversationGoals	**Field Type** ConversationDefinitionGoal[] on page 458 **Description** The list of goals in this bot template. Available in API version 57.0 and later.
conversationSystemDialogs	**Field Type** ConversationSystemDialog[] on page 459 **Description** A system function assigned to a dialog.
conversationVariables	**Field Type** ConversationVariable[] on page 460 **Description** A container that stores a specific piece of data collected from the customer. You can use variables within dialog actions as both inputs and outputs.
description	**Field Type** string **Description** A description of the bot template.
entryDialog	**Field Type** string

Field Name	Description
	Description A reference to the first dialog that the bot presents to your customer. For example, `Welcome`.
`icon`	**Field Type** string **Description** The icon used to identify the template.
`mainMenuDialog`	**Field Type** string **Description** A reference to the dialog identified as the main menu dialog. For example, `Main Menu`.
`masterLabel`	**Field Type** string **Description** Required. A user-friendly label for BotTemplate, which is defined when the BotTemplate is created.
`mlDomain`	**Field Type** LocalMlDomain on page 416 **Description** Required. Represents the Einstein Intent Set that groups intents, entities, and variables associated with a template.
`permissionSet`	**Field Type** string **Description** The permission set associated with the bot template. Available in API version 59.0 and later.
`richContentEnabled`	**Field Type** boolean **Description** Indicates whether the template is available for enhanced bots (`true`) or for standard bots (`false`). The default is `false`.
`type`	**Field Type** BotType (enumeration of type string)

Field Name	Description
	Description This field represents the configuration type of the bot. The default value is `Bot`. Valid values are: • `Bot`—Default Einstein Bot configuration. • `ExternalCopilot`— An external-facing agent. For example, an Agentforce Service agent. • `InternalCopilot`— An internal-facing agent. For example, an Agentforce Employee agent.

Declarative Metadata Sample Definition

The following is an example of a BotTemplate component.

```
<?xml version="1.0" encoding="UTF-8"?>
<BotTemplate xmlns="http://soap.sforce.com/2006/04/metadata">
    <botDialogGroups>
        <developerName>dialog_group1</developerName>
        <label>dialog group1</label>
    </botDialogGroups>
    <botDialogs>
        <developerName>Test_Dialog_1</developerName>
        <label>Test_Dialog_1</label>
        <showInFooterMenu>false</showInFooterMenu>
    </botDialogs>
    <botDialogs>
        <developerName>Test_Dialog_2</developerName>
        <label>Test_Dialog_2</label>
        <showInFooterMenu>false</showInFooterMenu>
    </botDialogs>
    <botDialogs>
        <botSteps>
            <botMessages>
                <message>Hi! I&apos;m your helpful bot.</message>
                <messageIdentifier>m1</messageIdentifier>
            </botMessages>
            <stepIdentifier>s1</stepIdentifier>
            <type>Message</type>
        </botSteps>
        <botSteps>
           <conversationRecordLookup>
               <SObjectType>Account</SObjectType>
               <conditions>
                   <leftOperand>Account.Phone</leftOperand>
                   <operatorType>Equal</operatorType>
                   <rightOperandValue>Value</rightOperandValue>
                   <sortOrder>0</sortOrder>
               </conditions>
               <lookupFields>
```

```
                    <fieldName>Account.Phone</fieldName>
                </lookupFields>
                <lookupFields>
                    <fieldName>Account.OwnerId</fieldName>
                </lookupFields>
                <maxLookupResults>1</maxLookupResults>
                <sourceVariableName>_LastCustomerInput</sourceVariableName>
                <sourceVariableType>ConversationVariable</sourceVariableType>
                <targetVariableName>MyCustomVariable</targetVariableName>
            </conversationRecordLookup>
            <stepIdentifier>s2</stepIdentifier>
            <type>RecordLookup</type>
        </botSteps>
        <botSteps>
            <botNavigation>
                <botNavigationLinks>
                    <targetBotDialog>Main_Menu</targetBotDialog>
                </botNavigationLinks>
                <type>Redirect</type>
            </botNavigation>
            <stepIdentifier>s3</stepIdentifier>
            <type>Navigation</type>
        </botSteps>
        <developerName>Welcome</developerName>
        <label>Welcome</label>
        <mlIntent>Welcome</mlIntent>
        <showInFooterMenu>false</showInFooterMenu>
    </botDialogs>
    <botDialogs>
        <botSteps>
            <stepIdentifier>s4</stepIdentifier>
            <type>Wait</type>
        </botSteps>
        <developerName>Main_Menu</developerName>
        <label>Main Menu</label>
        <mlIntent>Main_Menu</mlIntent>
        <showInFooterMenu>false</showInFooterMenu>
    </botDialogs>
    <botDialogs>
        <botSteps>
            <botMessages>
                <message>Goodbye! Click the &amp;quot;End Chat&amp;quot; button to end this
chat</message>
                <messageIdentifier>m2</messageIdentifier>
            </botMessages>
            <stepIdentifier>s6</stepIdentifier>
            <type>Message</type>
        </botSteps>
        <botSteps>
            <stepIdentifier>s7</stepIdentifier>
            <type>Wait</type>
        </botSteps>
        <developerName>End_Chat</developerName>
        <label>End Chat</label>
```

```
        <mlIntent>End_Chat</mlIntent>
        <showInFooterMenu>false</showInFooterMenu>
    </botDialogs>
    <botDialogs>
        <botSteps>
            <botMessages>
              <message>Unfortunately, there are no agents available at the moment</message>

                <messageIdentifier>m3</messageIdentifier>
            </botMessages>
            <stepIdentifier>s8</stepIdentifier>
            <type>Message</type>
        </botSteps>
        <botSteps>
            <stepIdentifier>s9</stepIdentifier>
            <type>Wait</type>
        </botSteps>
        <developerName>No_Agent_Available</developerName>
        <label>No Agent</label>
        <showInFooterMenu>false</showInFooterMenu>
    </botDialogs>
    <contextVariables>
        <contextVariableMappings>
            <SObjectType>LiveChatTranscript</SObjectType>
            <fieldName>LiveChatTranscript.ChatKey</fieldName>
            <messageType>WebChat</messageType>
        </contextVariableMappings>
        <dataType>Text</dataType>
        <developerName>ChatKey</developerName>
        <label>Chat Key</label>
    </contextVariables>
    <conversationLanguages>en_US</conversationLanguages>
    <conversationSystemDialogs>
        <dialog>No_Agent_Available</dialog>
        <type>TransferFailed</type>
    </conversationSystemDialogs>
    <conversationSystemDialogs>
        <dialog>Test_Dialog_1</dialog>
        <type>ErrorHandling</type>
    </conversationSystemDialogs>
    <conversationVariables>
        <dataType>Text</dataType>
        <developerName>TestVariableXYZ</developerName>
        <label>TestVariableXYZ</label>
    </conversationVariables>
    <conversationVariables>
        <collectionType>List</collectionType>
        <dataType>Object</dataType>
        <developerName>MyCustomVariable</developerName>
        <label>MyCustomVariable</label>
    </conversationVariables>
    <description>Description of BotTemplate</description>
    <entryDialog>Test_Dialog_1</entryDialog>
    <icon>AA8qwqXXXXX</icon>
```

```
    <mainMenuDialog>Test_Dialog_2</mainMenuDialog>
    <masterLabel>Astro Bot</masterLabel>
    <mlDomain>
        <label>Astro Bot</label>
        <mlIntents>
            <developerName>End_Chat</developerName>
            <label>End Chat</label>
            <mlIntentUtterances>
                <utterance>Utterance1</utterance>
                <language>es</language>
            </mlIntentUtterances>
            <mlIntentUtterances>
                <utterance>Utterance2</utterance>
                <language>es</language>
            </mlIntentUtterances>
            <mlIntentUtterances>
                <utterance>Utterance3</utterance>
                <language>es</language>
            </mlIntentUtterances>
        </mlIntents>
        <mlIntents>
            <developerName>Main_Menu</developerName>
            <label>Main Menu</label>
            <description>Main Menu Intent</description>
        </mlIntents>
        <mlIntents>
            <developerName>Welcome</developerName>
            <label>Welcome</label>
            <description>Welcome Intent</description>
        </mlIntents>
        <name>Astro_Bot_ld1</name>
    </mlDomain>
    <richContentEnabled>true</richContentEnabled>
</BotTemplate>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>BotTemplate</name>
    </types>
    <version>55.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

BotVersion

Represents the configuration details for a specific Einstein Bot version, including dialogs and variables.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

BotVersion components have the suffix `.bot` and are stored in the `bot` folder. BotVersion is a top-level child of Bot and shares its suffix and file directory.

Version

BotVersion components are available in API version 43.0 and later.

Special Access Rules

BotVersion is available only if Chat and Einstein Bots are enabled in your org.

Fields

Field Name	Description
`botDialogGroups`	**Field Type** BotDialogGroup[] on page 438 **Description** The list of dialog groups in this bot version.
`botDialogs`	**Field Type** BotDialog[] on page 439 **Description** The list of dialogs in this bot version.
`company`	**Field Type** string **Description** Reserved for internal use.
`conversationGoals`	**Field Type** ConversationDefinitionGoal[] on page 458 **Description** The list of goals in this bot verion. Available in API version 57.0 and later.

Field Name	Description
`conversationPlanner`	**Field Type** ConversationDefinitionPlanner[] on page 459 **Description** Represents the API name of the Agent planner service GenAiPlanner on page 1252. Available in API version 60.0 and later.
`conversationSystemDialogs`	**Field Type** ConversationSystemDialog[] on page 459 **Description** A system function assigned to a dialog. Available in API version 48.0 and later.
`conversationVariables`	**Field Type** ConversationVariable[] on page 460 **Description** A container that stores a specific piece of data collected from the customer. You can use variables within dialog actions as both inputs and outputs. Available in API version 44.0 and later.
`copilotPrimaryLangauge`	**Field Type** Language (enumeration of type string) **Description** Represents the primary language of a Copilot or Agent.
`copilotSecondaryLanguages`	**Field Type** string **Description** Reserved for internal use.
`entryDialog`	**Field Type** string **Description** Required. A reference to the first dialog that the bot presents to your customer. For example, `Welcome`.
`initialIntentDetectionEnabled`	**Field Type** boolean **Description** Reserved for internal use.
`intentDisambiguationEnabled`	**Field Type** boolean

Field Name	Description
	Description Reserved for internal use.
`intentThreshold`	**Field Type** double **Description** Specifies how strictly a user message must match with a bot intent. Valid values are between 1 and 5, where 1 is the least strict and 5 is the most strict. To turn on this feature, contact Salesforce Customer Support. This field is available in API version 63.0 and later.
`intentV3Enabled`	**Field Type** boolean **Description** Reserved for internal use.
`knowledgeActionEnabled`	**Field Type** boolean **Description** Indicates whether a knowledge action is enabled. The default value is `false`.
`knowledgeFallbackEnabled`	**Field Type** boolean **Description** Reserved for internal use.
`mainMenuDialog`	**Field Type** string **Description** A reference to the dialog identified as the main menu dialog. For example, `Main Menu`.
`nlpProviders`	**Field Type** ConversationDefinitionNlpProvider[] on page 461 **Description** Defines the language provider which is used for a particular language. Available in API version 49.0 and later.
`responseDelayMilliseconds`	**Field Type** int **Description** An optional default or custom delay after every bot response to simulate typing.

Field Name	Description
`role`	**Field Type** string **Description** Reserved for internal use.
`surfacesEnabled`	**Field Type** boolean **Description** Reserved for internal use.
`toneType`	**Field Type** GenAiBotToneType (enumeration of type string) **Description** The tone of the bot. Valid values are: • `Casual` • `Formal` • `Neutral`

BotDialogGroup

The list of dialog groups in this bot version.

Field Name	Description
`description`	**Field Type** string **Description** A description of the bot dialog group.
`developerName`	**Field Type** string **Description** Required. This unique name prevents conflicts with other dialog groups associated with the same bot version. This name can contain only underscores and alphanumeric characters. The name must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. Note: Only users with View DeveloperName OR View Setup and Configuration permission can view, group, sort, and filter this field.

Field Name	Description
`label`	**Field Type** string **Description** Required. A label that identifies the dialog group throughout the Salesforce user interface.

BotDialog

The list of dialogs in this bot version.

Field Name	Description
`botDialogGroup`	**Field Type** string **Description** The bot dialog group that contains this bot dialog.
`botSteps`	**Field Type** BotStep[] on page 440 **Description** A list of steps that are executed as part of the dialog.
`description`	**Field Type** string **Description** A description of the bot dialog.
`developerName`	**Field Type** string **Description** Required. This unique name prevents conflicts with other dialogs associated with the same bot version. This name can contain only underscores and alphanumeric characters. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. Note: Only users with View DeveloperName OR View Setup and Configuration permission can view, group, sort, and filter this field.
`isPlaceholderDialog`	**Field Type** boolean

Field Name	Description
	Description In a bot block, indicates whether a dialog is a placeholder (`true`) or not (`false`). In a bot template or bot version not associated with a bot block, this field is read-only and the value is `false`. Available in API version 58.0 and later.
`label`	**Field Type** string **Description** Required. A label that identifies the dialog throughout the Salesforce user interface.
`mlIntent`	**Field Type** string **Description** Required. A label that identifies the dialog throughout the Salesforce user interface. The name of the intent associated with a dialog.
`mlIntentTrainingEnabled`	**Field Type** boolean **Description** Indicates whether Einstein is turned on to train an intent model for the dialog intent (`true`) or turned off for the exact match option (`false`). The default value is `false`. Available in API version 46.0 and later.
`showInFooterMenu`	**Field Type** boolean **Description** Indicates whether to show this dialog in the Bot Options menu. The default value is `false`.

BotStep

A step that is executed as part of the dialog.

Field Name	Description
`booleanFilter`	**Field Type** string **Description** This field is reserved for future use.

Field Name	Description
`botInvocation`	**Field Type** BotInvocation on page 443 **Description** Bot Invocation used by a BotStep of type `Invocation`.
`botMessages`	**Field Type** BotMessage[] on page 445 **Description** List of bot messages used by a BotStep of type `Message`.
`botNavigation`	**Field Type** BotNavigation on page 445 **Description** Bot Navigation used by a BotStep of type `Navigation`.
`botStepConditions`	**Field Type** BotStepCondition[] on page 446 **Description** List of BotStep conditions associated with a BotStep of type `Group`.
`botSteps`	**Field Type** BotStep[] on page 440 **Description** List of BotSteps associated to a Bot Step of type `Group`.
`botVariableOperation`	**Field Type** BotVariableOperation[] on page 447 **Description** Bot Variable Operation used by a BotStep of type `VariableOperation`.
`conditionLogicType`	**Field Type** ConversationDefinitionLogicalOperatorType (enumeration of type string) **Description** Represents the type of conditional logic used by a BotStep. Values are: • `And` • `Or` Available in API version 58.0 and later.
`conversationRecordLookup`	**Field Type** ConversationRecordLookup[] on page 453

Field Name	Description
	Description A lookup action to the Conversation record. Available in API version 46.0 and later.
`conversationStepGoalMappings`	**Field Type** ConversationDefinitionStepGoalMapping[] on page 456 **Description** The API name of a goal used by a BotStep of type GoalStep. Available in API version 57.0 and later.
`conversationSystemMessage`	**Field Type** ConversationSystemMessage[] on page 456 **Description** System messages that represent an action for a BotStep, such as transferring to an agent or ending a chat. Available in API version 46.0 and later.
`messageDefinition`	**Field Type** ConversationDefinitionRichMessage[] on page 458 **Description** List of configuration details used by a BotStep that references a messaging component. Available in API version 54.0 and later.
`stepIdentifier`	**Field Type** string **Description** A unique key that identifies a step within a dialog. It is used to link translated labels to labels within the step. This field is recommended for all step records and is required for translated step labels. Available in API version 53.0 and later. If a step is created via the UI, the `stepIdentifier` is automatically generated. If a step is created via API, the `stepIdentifier` must be provided. The `stepIdentifier` can contain letters, numbers, dashes, and underscores, up to 255 characters.
`type`	**Field Type** BotStepType (enumeration of type string) **Description** Required. Values are: • `GoalStep` (Available in API version 57.0 and later.) • `Group` • `Invocation` • `Message`

Field Name	Description
	• `Navigation` • `RecordLookup` (Available in API version 48.0 and later.) • `RichMessage` (Available in API version 54.0 and later.) • `SystemMessage` • `VariableOperation` • `Wait`

BotInvocation

Bot Invocation used by a BotStep of type `Invocation`.

Field Name	Description
`invocationActionName`	**Field Type** string **Description** The name of the invocable action used by a Bot Invocation.
`invocationActionType`	**Field Type** ConversationInvocableTargetType (enumeration of type string) **Description** Available dialog action types are: Values are: • `apex` • `api` (Available in API version 64.0 and later.) • `botAction` (Available in API version 64.0 and later.) • `externalService` (Available in API version 53.0 and later.) • `flow` • `logFeedback` (Available in API version 51.0 and later.) • `logGoalAchieved` (Deprecated in API version 57.0 and later.) • `standardInvocableAction`
`invocationMappings`	**Field Type** BotInvocationMapping[] on page 443 **Description** List of Bot Invocation Mappings for a Bot Invocation.

BotInvocationMapping

List of Bot Invocation Mappings for a Bot Invocation.

Field Name	Description
`parameterName`	**Field Type** string **Description** Required. Name of an Input/Output parameter of the parent Bot Invocation target.
`recordName`	**Field Type** string **Description** Name of the record that is used as part of an Invocation mapping. Available in API version 54.0 and later.
`type`	**Field Type** BotInvocationMappingType (enumeration of type string) **Description** Required. Values are: • `Input` • `Output`
`value`	**Field Type** string **Description** Literal value to be assigned to the specified parameter.
`variableName`	**Field Type** string **Description** Name of the Bot Variable that is used as part of an Invocation mapping.
`variableType`	**Field Type** ConversationVariableType (enumeration of type string) **Description** This field relates to the type of variable used in this invocation mapping. Values are: • `ContextVariable` • `ConversationVariable` • `PageContextVariable`

BotMessage

A bot message used by a BotStep of type `Message`.

Field Name	Description
`message`	**Field Type** string **Description** Required. Message to display as part of an outgoing message from the bot to the customer.
`messageIdentifier`	**Field Type** string **Description** A unique key that identifies a message within a dialog. It is used to link translated labels to labels within the message. This field is recommended for all message records and is required for translated message labels. Available in API version 53.0 and later. If a message is created via the UI, the `messageIdentifier` is automatically generated. If a message is created via API, the `messageIdentifier` must be provided. `messageIdentifier` can contain letters, numbers, dashes, and underscores, up to 255 characters.

BotNavigation

Bot navigation used by a BotStep of type `Navigation`.

Field Name	Description
`botNavigationLinks`	**Field Type** BotNavigationLink[] on page 446 **Description** List of Bot Navigation links associated with a Bot Navigation of type `Call` or `Redirect`.
`type`	**Field Type** BotNavigationType (enumeration of type string) **Description** Required. Values are: • `Call` • `Redirect` • `TransferToAgent`

BotNavigationLink

List of Bot Navigation links associated with a Bot Navigation of type `Call` or `Redirect`.

Field Name	Description
`label`	**Field Type** string **Description** Label displayed when more than one Bot Navigation Link is available under a Bot Navigation of type `Redirect`. The target dialog label is used when no label is provided.
`targetBotDialog`	**Field Type** string **Description** Name of the target dialog to be called as part of this Bot Navigation Link.
`targetVariable`	**Field Type** string **Description** In the Redirect to Dialog Rule Action, the ID of the target object variable to be called as part of this Bot Navigation link. Available in API version 57.0 and later.
`targetVariableType`	**Field Type** ConversationVariableType (enumeration of type string) **Description** In the Redirect to Dialog Rule Action, the type of variable referred to in `targetVariable`. Available in API version 57.0 and later. Values are: • `ContextVariable` • `ConversationVariable` • `PageContextVariable`

BotStepCondition

List of BotStep conditions associated with a BotStep of type `Group`.

Field Name	Description
`leftOperandName`	**Field Type** string **Description** Required.

Field Name	Description
	Name of the variable used as the left side of the condition operation.
`leftOperandType`	**Field Type** ConversationVariableType (enumeration of type string) **Description** Required. Type of the variable used as the left side of the condition operation. Values are: • `ContextVariable` • `ConversationVariable` • `PageContextVariable`
`operatorType`	**Field Type** BotStepConditionOperatorType (enumeration of type string) **Description** Required. Values are: • `Equals` • `GreaterThan` (Available in API version 47.0 and later.) • `GreaterThanOrEqualTo` (Available in API version 47.0 and later.) • `IsNotSet` • `IsSet` • `LessThan` (Available in API version 47.0 and later.) • `LessThanOrEqualTo` (Available in API version 47.0 and later.) • `NotEquals`
`rightOperandValue`	**Field Type** string **Description** Value that is used as the right side of the condition operation. This value is ignored when using `IsSet` and `IsNotSet` operators.

BotVariableOperation

Bot variable operation used by a BotStep of type `VariableOperation`.

Field Name	Description
`askCollectIfSet`	**Field Type** boolean

Field Name	Description
	Description If `true`, the bot runs a Bot Variable Operation of type `Collect` regardless of whether the variable already has a value. When a value exists for a variable, the bot asks the user for the relevant information, and the bot overwrites the existing value with the user-provided value. If `false`, the bot skips variables with an existing value and maintains the existing value. The default is `false`. Available in API version 51.0 and later.
`autoSelectIfSingleChoice`	**Field Type** boolean **Description** If `true`, the bot automatically selects the answer in the conversation flow when only one button choice is available in a Bot Variable Operation of type `Collect` and a `quickReplyType` value of `Dynamic`. If `false`, the bot presents the single button choice and waits for the user's response. The default is `false`. Available in API version 51.0 and later.
`botInvocation`	**Field Type** BotInvocation on page 443 **Description** Bot Invocation used to provide Dynamic choices by a Bot Variable Operation of type `Collect` and `quickReplyType` of `Dynamic`.
`botMessages`	**Field Type** BotMessage[] on page 445 **Description** List of Bot Messages used as prompt messages by a Bot Variable Operation of type `Collect`.
`botQuickReplyOptions`	**Field Type** BotQuickReplyOption[] on page 451 **Description** List of static choice options used by a Bot Variable Operation of type `Collect` and `quickReplyType` of `Static`.
`botVariableOperands`	**Field Type** BotVariableOperand[] on page 451 **Description** List of Bot Variable Operands associated with a Bot Variable of type `Set` or `Unset`.
`ignoreIntentRecognition`	**Field Type** boolean

Field Name	Description
	Description If `true` the bot requires a response to a Question dialog step. The bot doesn't perform intent recognition for any user messages that do not fit the entity requirements. The bot repeats the question until the customer's response fits the entity requirements.
`invalidInputBotNavigation`	**Field Type** BotNavigation on page 445 **Description** Bot Navigation used by a Bot Variable Operation of type `Collect`. This navigation is executed when the associated Bot Invocation doesn't return any options.
messageDefinition	**Field Type** ConversationDefinitionRichMessage on page 458 **Description** Configuration details that reference a messaging component. Outputs are used by a Bot Variable Operation of type `Set`. Available in API version 58.0 and later.
`optionalCollect`	**Field Type** boolean **Description** If `true`, the bot asks the repair attempts once and then moves on to the next dialog step. The default value is `false`. Available in API version 48.0 and later.
`quickReplyOptionTemplate`	**Field Type** string **Description** Formula template used to resolve a label for Dynamic choice options of type `Object`.
`quickReplyType`	**Field Type** BotQuickReplyType (enumeration of type string) **Description** Values are: • `Dynamic` • `Static`
`quickReplyWidgetType`	**Field Type** BotWidgetType (enumeration of type string) **Description** Values are: • `Buttons` • `Menu`

Field Name	Description
`retryMessages`	**Field Type** BotMessage[] on page 445 **Description** In Conversation Repair, the messages assigned to repair attempts. Available in API version 48.0 and later.
`sourceVariableName`	**Field Type** string **Description** Name of the source `VariableName` used in the variable operation. Available in API version 47.0 and later.
`sourceVariableType`	**Field Type** ConversationVariableType (enumeration of type string) **Description** This name defines the data type of `VariableName` used in the variable operation. Values are: • `ContextVariable` • `ConversationVariable` • `PageContextVariable`
`successMessages`	**Field Type** BotMessage[] on page 445 **Description** In a File dialog step, the message displayed to the customer as part of type `CollectAttachment` to confirm a successful file upload. Available in API version 57.0 and later.
`type`	**Field Type** BotVariableOperationType (enumeration of type string) **Description** Required. Values are: • `Collect` • `CollectAttachment` (Available in API version 57.0 and later.) • `Set` • `SetConversationLanguage` (Available in API version 53.0 and later.) • `Unset`
`variableOperationIdentifier`	**Field Type** string

Field Name	Description
	Description A unique key that identifies a variable operation within a dialog. It is used to link translated labels to labels within the variable operation. This field is recommended for all variable operation records and is required for translated variable operation labels. Available in API version 53.0 and later. If a variable operation is created via the UI, the `variableOperationIdentifier` is automatically generated. If a variable operation is created via API, the `variableOperationIdentifier` must be provided. `variableOperationIdentifier` can contain letters, numbers, dashes, and underscores, up to 255 characters.

BotQuickReplyOption

List of static choice options used by a bot variable operation of type `Collect` and `quickReplyType` of `Static`.

Field Name	Description
`literalValue`	**Field Type** string **Description** Required. Value to be displayed as a menu or button choice to your customer.
`quickReplyOptionIdentifier`	**Field Type** string **Description** A unique key that identifies a quick reply option within a dialog. It is used to link translated labels to labels within the quick reply option. This field is recommended for all quick reply option records and is required for translated quick reply option labels. Available in API version 53.0 and later. If a quick reply option is created via the UI, the `quickReplyOptionIdentifier` is automatically generated. If a message is created via API, the `quickReplyOptionIdentifier` must be provided. `quickReplyOptionIdentifier` can contain letters, numbers, dashes, and underscores, up to 255 characters.

BotVariableOperand

List of bot variable operands associated with a bot variable of type `Set` or `Unset`.

Field Name	Description
`disableAutoFill`	**Field Type** boolean **Description** Disables auto-fill behavior for a bot variable under a bot variable operation of type `Collect`.
`sourceName`	**Field Type** string **Description** Name of the source CustomField or MlSlotClass used in the variable operation.
`sourceType`	**Field Type** ConversationVariableOperandSourceType (enumeration of type string) **Description** Values are: • `BotDefinition` (Available in API version 46.0 and later.) • `ContextVariable` (Available in API version 45.0 and later.) • `ConversationVariable` • `FlowDefinition` (Available in API version 52.0 and later.) • `MlSlotClass` • `Queue` (Available in API version 46.0 and later.) • `StandardConversationVariable` • `StandardMlSlotClass` • `Value`
`sourceValue`	**Field Type** string **Description** Literal value used as the source for this variable operation.
`targetName`	**Field Type** string **Description** Required. Name of the target variable used in the variable operation.
`targetType`	**Field Type** ConversationVariableType (enumeration of type string) **Description** Required.

Field Name	Description
	Type of the target used in the variable operation. Values are: • `ContextVariable` • `ConversationVariable` • `PageContextVariable`

ConversationRecordLookup

Information related to the linked conversation. Currently only works on Lightning Knowledge. Available in API version 46.0 and later.

Field Name	Description
`SObjectType`	**Field Type** string **Description** Required. Specifies the SObjectType of the ID stored in a bot variable.
`conditions`	**Field Type** ConversationRecordLookupCondition[] on page 455 **Description** The conditions associated with this lookup. Available in API version 51.0 and later.
`filterLogic`	**Field Type** string **Description** The logical operator that connects the conditions. Values are: • `And` • `Or` Available in API version 51.0 and later.
`lookupFields`	**Field Type** ConversationRecordLookupField[] on page 456 **Description** Definition of the fields that are used for this lookup.
`maxLookupResults`	**Field Type** int

Field Name	Description
	Description Required. The maximum number of records to return (1-3).
`sortFieldName`	**Field Type** string **Description** The name of the field used to sort the lookup results. Available in API version 51.0 and later.
`sortOrder`	**Field Type** SortOrder (enumeration of type string) **Description** The display order of the lookup results. Values are: • `Asc` • `Desc` Available in API version 51.0 and later.
`sourceVariableName`	**Field Type** string **Description** The input for this lookup operation.
`sourceVariableType`	**Field Type** ConversationVariableType (enumeration of type string) **Description** Type of the target used in the variable operation. Values are: • `ContextVariable` • `ConversationVariable` • `PageContextVariable`
`targetVariableName`	**Field Type** string **Description** Required. The variable that holds the results of this lookup.

ConversationRecordLookupCondition

List of conditions associated with a ConversationRecordLookup. Available in API version 51.0 and later.

Field Name	Description
`leftOperand`	**Field Type** string **Description** Required. Field on which the condition operation takes place.
`operatorType`	**Field Type** string **Description** Required. The operator applied to the leftOperand. Values are: • `Equals` • `NotEquals` • `IsSet` • `IsNotSet` • `GreaterThan` • `LessThan` • `GreaterThanOrEqualTo` • `LessThanOrEqualTo`
`rightOperandName`	**Field Type** string **Description** The name of the variable to compare against.
`rightOperandType`	**Field Type** ConversationVariableType (enumeration of type string) **Description** The type of the variable to compare against. Values are: • `ContextVariable` • `ConversationVariable` • `PageContextVariable`

Field Name	Description
`rightOperandValue`	**Field Type** string **Description** The custom value to compare against. This value is ignored when using `IsSet` and `IsNotSet` operators.
`sortOrder`	**Field Type** int **Description** Required. Order in which the conditions are applied.

ConversationRecordLookupField

The fields used in a conversation record lookup. Available in API version 46.0 and later.

Field Name	Description
`fieldName`	**Field Type** string **Description** Required. Defines the field names used in the Conversation Lookup function.

ConversationDefinitionStepGoalMapping

Represents the association between a goal and a BotStep. A goal can be associated with only one BotStep and one dialog at a time. Available in API version 57.0 and later.

Field Name	Description
`goalName`	**Field Type** string **Description** The API name of the goal.

ConversationSystemMessage

System messages that represent an action for a Bot Step, such as transferring to an agent or ending a chat. Available in API version 46.0 and later.

Field Name	Description
`systemMessageMappings`	**Field Type** ConversationSystemMessageMapping on page 457 **Description** Defines the type of system message to be sent.
`type`	**Field Type** ConversationSystemMessageType (enumeration of type string) **Description** Required. This field defines the values available for a system message. Values are: • `EndChat` • `Transfer`

ConversationSystemMessageMapping

List of mappings that indicate additional information provided for the system message. Available in API version 46.0 and later.

Field Name	Description
`mappingType`	**Field Type** ConversationMappingType (enumeration of type string) **Description** Required. Defines the type of mapping used in the record. Values are: • `Input` • `Output`
`parameterType`	**Field Type** ConversationSystemMessageParamType (enumeration of type string) **Description** Required. Defines the type of parameter the value is mapped to. Values are: • `Transfer`
`variableName`	**Field Type** string

Field Name	Description
	Description Required. Name of the variable that contains the value passed to the system message.

ConversationDefinitionRichMessage

Represents the configuration details for referencing a messaging component, such as an enhanced link. Available in API version 54.0 and later.

Field Name	Description
`messageDefinitionMappings`	**Field Type** BotInvocationMapping[] on page 443 **Description** List of mappings for referencing a messaging component. Includes any input parameters and their values. Optionally, specifies the conversation variable for storing any outputs. Input parameter values can be either static values or references to conversation or context variables.
`messageDefinitionName`	**Field Type** string **Description** Required. The API name of the messaging component referenced by the bot.

ConversationDefinitionGoal

A goal included in the bot version. Available in API version 57.0 and later.

Field Name	Description
`developerName`	**Field Type** string **Description** Required. A unique name that prevents conflicts with other goals associated with the same bot version. This name can contain only underscores and alphanumeric characters. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.

Field Name	Description
`label`	**Field Type** string **Description** Required. A label that identifies the goal throughout the Salesforce user interface. This label can contain only underscores and alphanumeric characters. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.

ConversationDefinitionPlanner

Represents the API name for the Agent planner service. Available in API version 60.0 and later.

Field Name	Description
`genAiPlannerName`	**Field Type** string **Description** The name of an agent planner service that uses a large language model (LLM) and a reasoning strategy to decompose a given task into smaller subtasks, identify the most suitable actions for each subtask, and invoke them.

ConversationSystemDialog

A system function assigned to a dialog. Available in API version 48.0 and later.

Field Name	Description
`dialog`	**Field Type** string **Description** The dialog name triggered when this system event fires.
`type`	**Field Type** ConversationSystemDialogType (enumeration of type string) **Description** The type of system event. Required. Valid values are: • `Disambiguation` (Reserved for Future Use) • `DisambiguationFailed` (Reserved for Future Use) • `ErrorHandling` • `KnowledgeAction` (Available in API version 60.0.)

Field Name	Description
	• `KnowledgeFallback` (Available in API version 51.0.) • `TransferFailed`

ConversationVariable

A container that stores a specific piece of data collected from the customer. You can use variables within dialog actions as both inputs and outputs. Available in API version 44.0 and later.

Field Name	Description
`collectionType`	**Field Type** ConversationVariableCollectionType (enumeration of type string) **Description** This field defines whether a variable is designated as a List Variable. Valid value is List.
`dataType`	**Field Type** ConversationVariableCollectionType (enumeration of type string) **Description** Required. Valid values are: • `Boolean` • `Currency` • `Date` • `DateTime` • `Id` (available in API 45.0 and later.) • `Object` • `Number` • `Text`
`description`	**Field Type** string **Description** A description of this variable. This value may be used by the Agentforce planner service. Available in API version 63.0 and later.
`developerName`	**Field Type** string **Description** Required.

Field Name	Description
	This name can contain only underscores and alphanumeric characters and must be unique in your org. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. Only users with View DeveloperName OR View Setup and Configuration permission can view, group, sort, and filter this field.
`includeInPrompt`	**Field Type** boolean **Description** Indicates whether the variable is injected into the prompt sent to the Agentforce model. Available in API version 63.0 and later.
`label`	**Field Type** string **Description** Required. Label that identifies a variable throughout the Salesforce user interface.
`SObjectType`	**Field Type** string **Description** Specifies the SObjectType of the ID stored in a bot variable. Valid values are: • `BotDefinition` • `Queue`
`visibility`	**Field Type** ConversationVariableVisibilityType (enumeration of type string) **Description** Required. Determines which components can set this variable. If the visibility is `internal`, the variable can only be set by action outputs. If the visibility is `external`, the variable can also be set by the API. Valid values are: • `internal` • `external`

ConversationDefinitionNlpProvider

Defines the natural language service that is used for the language assigned to a bot version. Available in API version 49.0 and later.

Field Name	Description
`language`	**Field Type** Language **Description** Required. The language assigned to a bot version.
`nlpProviderName`	**Field Type** string **Description** If nlpProviderType is EinsteinAI, this field is blank. If Apex, this field holds the Apex class name of the service.
`nlpProviderType`	**Field Type** ConversationDefinitionNlpProviderType (enumeration of type string) **Description** Required. Default value is `EinsteinAi`. Valid values are: • `EinsteinAi` • `Apex`

Declarative Metadata Sample Definition

The following is an example of a BotVersion.

```
<?xml version="1.0" encoding="UTF-8"?>
<Bot xmlns="http://soap.sforce.com/2006/04/metadata">
    <botMlDomain>
        <label>Astros Pizza</label>
        <mlIntents>
            <developerName>New_Order</developerName>
            <label>New Order</label>
            <mlIntentUtterances>
                <utterance>Today is pie day so I want pie</utterance>
            </mlIntentUtterances>
        </mlIntents>
        <mlSlotClasses>
            <developerName>Size</developerName>
            <extractionType>Value</extractionType>
            <label>Size</label>
            <mlSlotClassValues>
                <synonymGroup>
                    <languages>en_US</languages>
                    <terms>Big</terms>
                    <terms>Extra Large</terms>
```

```
                <terms>X-Large</terms>
                <terms>Grande</terms>
                <terms>Huge</terms>
            </synonymGroup>
            <value>Large</value>
        </mlSlotClassValues>
    </mlSlotClasses>
    <name>Astros_Pizza_ld1</name>
</botMlDomain>
<botVersions>
    <fullName>v1</fullName>
    <botDialogGroups>
        <developerName>Order_Management</developerName>
        <label>Order Management</label>
    </botDialogGroups>
    <botDialogs>
        <botDialogGroup>Order_Management</botDialogGroup>
        <botSteps>
            <botMessages>
                <message>□□□□□□Pizza Time! □□□□□□</message>
                <messageIdentifier>Greeting_Message</messageIdentifier>
            </botMessages>
            <stepIdentifier>Greeting</stepIdentifier>
            <type>Message</type>
        </botSteps>
        <botSteps>
            <botStepConditions>
                <leftOperandName>Verified_User</leftOperandName>
                <leftOperandType>ConversationVariable</leftOperandType>
                <operatorType>Equals</operatorType>
                <rightOperandValue>false</rightOperandValue>
            </botStepConditions>
            <botSteps>
                <botNavigation>
                    <botNavigationLinks>
                        <targetBotDialog>Customer_Verification</targetBotDialog>
                    </botNavigationLinks>
                    <type>Call</type>
                </botNavigation>
                <stepIdentifier>Call_Customer_Verification</stepIdentifier>
                <type>Navigation</type>
            </botSteps>
            <stepIdentifier>Verify_User</stepIdentifier>
            <type>Group</type>
        </botSteps>
        <botSteps>
            <botStepConditions>
                <leftOperandName>Location</leftOperandName>
                <leftOperandType>ConversationVariable</leftOperandType>
                <operatorType>IsNotSet</operatorType>
            </botStepConditions>
            <botSteps>
                <botNavigation>
                    <botNavigationLinks>
```

```
                        <targetBotDialog>Select_Location</targetBotDialog>
                    </botNavigationLinks>
                    <type>Call</type>
                </botNavigation>
                <stepIdentifier>Call_Select_Location</stepIdentifier>
                <type>Navigation</type>
            </botSteps>
            <stepIdentifier>Set_Location</stepIdentifier>
            <type>Group</type>
        </botSteps>
        <botSteps>
            <botVariableOperation>
                <botInvocation>
                    <invocationActionName>CreateOrderService</invocationActionName>
                    <invocationActionType>apex</invocationActionType>
                    <invocationMappings>
                        <parameterName>customer</parameterName>
                        <type>Input</type>
                        <variableName>Contact</variableName>
                        <variableType>ConversationVariable</variableType>
                    </invocationMappings>
                    <invocationMappings>
                        <parameterName>location</parameterName>
                        <type>Input</type>
                        <variableName>Location</variableName>
                        <variableType>ConversationVariable</variableType>
                    </invocationMappings>
                    <invocationMappings>
                        <parameterName>output</parameterName>
                        <type>Output</type>
                        <variableName>Pizza_Order</variableName>
                        <variableType>ConversationVariable</variableType>
                    </invocationMappings>
                </botInvocation>
                <type>Set</type>
                <variableOperationIdentifier>Set_Order</variableOperationIdentifier>
            </botVariableOperation>
            <stepIdentifier>Create_Order</stepIdentifier>
            <type>VariableOperation</type>
        </botSteps>
        <botSteps>
            <botMessages>
             <message>Perfect, let&apos;s work on your order from our {!Location.Name}
location</message>
                <messageIdentifier>Start_Order_Message</messageIdentifier>
            </botMessages>
            <stepIdentifier>Start_Order</stepIdentifier>
            <type>Message</type>
        </botSteps>
        <botSteps>
            <messageDefinition>
                <messageDefinitionName>Astros_Pizza_Menu</messageDefinitionName>
            </messageDefinition>
            <stepIdentifier>36e5a7cb-50c4-4279-aa06-1217eba1bf62</stepIdentifier>
```

```
                <type>RichMessage</type>
            </botSteps>
            <botSteps>
                <botNavigation>
                    <botNavigationLinks>
                        <targetBotDialog>Add_Items_to_Order</targetBotDialog>
                    </botNavigationLinks>
                    <type>Redirect</type>
                </botNavigation>
                <stepIdentifier>Proceed_To_Add_Items</stepIdentifier>
                <type>Navigation</type>
            </botSteps>
            <developerName>New_Order</developerName>
            <label>New Order</label>
            <mlIntent>New_Order</mlIntent>
            <showInFooterMenu>false</showInFooterMenu>
        </botDialogs>
        <conversationVariables>
            <dataType>Object</dataType>
            <developerName>Contact</developerName>
            <label>Contact</label>
        </conversationVariables>
        <conversationVariables>
            <dataType>Text</dataType>
            <developerName>Delivery_Address</developerName>
            <label>Delivery Address</label>
        </conversationVariables>
        <conversationVariables>
            <dataType>Object</dataType>
            <developerName>Pizza_Order</developerName>
            <label>Pizza Order</label>
        </conversationVariables>
        <entryDialog>Welcome</entryDialog>
        <mainMenuDialog>Main_Menu</mainMenuDialog>
    </botVersions>
    <label>Astro&amp;apos;s Pizza</label>
</Bot>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>Astros Pizza_Bot.v1</members>
        <name>BotVersion</name>
    </types>
    <version>45.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type doesn't support the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

BrandingSet

Represents the definition of a set of branding properties for an Experience Builder site or for your org's Lightning Experience theme.

This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

BrandingSet components have the suffix `brandingSet` and are stored in the `brandingSets` folder.

Version

BrandingSet components are available in API version 40.0 and later.

Special Access Rules

The BrandingSet type is available when at least one of the following is enabled in your org: Digital Experiences, Surveys, or Lightning Experience. All users, including unauthenticated guest users, can access this type.

Fields

Field Name	Field Type	Description
`brandingSetProperty`	BrandingSetProperty[]	An array containing the name and value of each branding property, such as `TextColor:#333`.
`description`	string	A description of the set of branding properties.
`masterLabel`	string	Required. The user interface name of the set of branding properties.
`type`	string	The assigned branding set definition for this BrandingSet.

BrandingSetProperty

Represents the definition of a branding property in the Theme panel in Experience Builder or in the Edit Theme page in Setup.

Field Name	Field Type	Description
`propertyName`	string	Required. The name of the branding property, such as `TextColor`.
`propertyValue`	string	The value of the branding property, such as `#333`.

Branding Properties for Lightning Experience Themes

Use these properties to describe Lightning Experience themes. Each propertyName is case-sensitive and must use all capital letters. Some properties only apply to either Salesforce Lightning Design System (SLDS) 1 themes or SLDS 2 themes.

Note: SLDS 2 themes is a pilot or beta service that is subject to the Beta Services Terms at Agreements - Salesforce.com or a written Unified Pilot Agreement if executed by Customer, and applicable terms in the Product Terms Directory. Use of this pilot or beta service is at the Customer's sole discretion.

Property	Description
`ACCENT_COLOR_1`	**Field Type** string **Description** The primary accent color used to highlight active states of the user interface like navigation, tabs, and hover states. Must be a valid hex color string in the format #54C254. Available only for SLDS 2 themes. Available in API version 64.0 and later.
`ACCENT_COLOR_2`	**Field Type** string **Description** A variant of the primary accent color used to highlight active states of the user interface like navigation, tabs, and hover states. Must be a valid hex color string in the format #54C254. Available only for SLDS 2 themes. Available in API version 64.0 and later.
`ACCENT_COLOR_3`	**Field Type** string **Description** A variant of the primary accent color used to highlight active states of the user interface, like navigation, tabs, and hover states. Must be a valid hex color string in the format #54C254. Available only for SLDS 2 themes. Available in API version 64.0 and later.
`BANNER_IMAGE`	**Field Type** string **Description** The path to the image to display in the background of your org's pages. Use a JPG, PNG, or GIF that's 1800x360 pixels and smaller than 5 MB. Must refer to an asset file that already exists within the org. Available only for SLDS 1 themes.

Property	Description
BORDER_ACCENT_COLOR_1	**Field Type** string **Description** The primary color used for border colors for branded containers like outline buttons, navigation, and tabs. Must be a valid hex color string in the format #54C254. Available only for SLDS 2 themes. Available in API version 64.0 and later.
BORDER_ACCENT_COLOR_2	**Field Type** string **Description** A variant of the primary color used for border colors for branded containers like outline buttons, navigation, and tabs. Must be a valid hex color string in the format #54C254. Available only for SLDS 2 themes. Available in API version 64.0 and later.
BORDER_ACCENT_COLOR_3	**Field Type** string **Description** A variant of the primary color used for border colors for branded containers like outline buttons, navigation, and tabs. Must be a valid hex color string in the format #54C254. Available only for SLDS 2 themes. Available in API version 64.0 and later.
BRAND_COLOR	**Field Type** string **Description** The color to display on your nav bar and other important areas of Salesforce. Must be a valid hex color string in the format #54C254.
BRAND_IMAGE	**Field Type** string **Description** The path to the image to display as your logo. Use a JPG, PNG, or GIF that's 600x120 pixels and smaller than 5 MB. Must refer to an asset file that already exists within the org.
CONTAINER_ACCENT_COLOR_1	**Field Type** string

Property	Description
	Description The primary color used for the background of branded component containers like the brand button. Container accent colors are also used for hover states for branded component containers. Must be a valid hex color string in the format #54C254. Available only for SLDS 2 themes. Available in API version 64.0 and later.
`CONTAINER_ACCENT_COLOR_2`	**Field Type** string **Description** A variant of the primary color used for the background of branded component containers. Container accent colors are also used for hover states for branded component containers. Must be a valid hex color string in the format #54C254. Available only for SLDS 2 themes. Available in API version 64.0 and later.
`CONTAINER_ACCENT_COLOR_3`	**Field Type** string **Description** A variant of the primary color used for the background of branded component containers. Container accent colors are also used for hover states for branded component containers. Must be a valid hex color string in the format #54C254. Available only for SLDS 2 themes. Available in API version 64.0 and later.
`GROUP_IMAGE`	**Field Type** string **Description** The default group avatar image. Use a JPG, PNG, or GIF that's 200x200 pixels and smaller than 5 MB. Group owners can change their avatar image. Must refer to an asset file that already exists within the org.
`GROUPS_BANNER_IMAGE`	**Field Type** string **Description** The default banner image for group pages. Use a JPG, PNG, or GIF that's 1800x360 pixels and smaller than 5 MB. Group owners

Property	Description
	can change their banner image. Must refer to an asset file that already exists within the org.
HEADER_BACKGROUND_COLOR	**Field Type** string **Description** The color to display at the top of your org pages. Your logo, global search, and global actions appear on top of the global header background. Must be a valid hex color string in the format #54C254.
LINK_AS_BACKGROUND	**Field Type** boolean **Description** Indicates whether you want links in your org to use your selected brand color (`true`) or not (`false`). The default value is `true`. Available only for SLDS 1 themes.
OVERRIDE_A11Y_COLOR	**Field Type** string **Description** When you select a value for `BRAND_COLOR`, a color palette that complements your brand color and is WCAG 2.0 compliant is automatically generated. In places where your selected brand color isn't accessible, an accessible color is used instead. If you provide a value for `OVERRIDE_A11Y_COLOR`, your value is used in the instances described in place of an automatically generated color. Overriding the accessible brand color only updates the first tile in your brand-based color palette, which affects colors like links and buttons. Overriding the accessibility brand color can make text harder to read. Must be a valid hex color string in the format #54C254. Available only for SLDS 1 themes.
OVERRIDE_LOADING_PAGE	**Field Type** boolean **Description** Indicates whether your provided brand logo displays while a Lighting Experience page loads or refreshes (`true`) or not (`false`). The default value is `false`.

Property	Description
PAGE_BACKGROUND_COLOR	**Field Type** string **Description** The color used for page backgrounds. Must be a valid hex color string in the format #54C254. Available only for SLDS 1 themes.
PROFILE_BANNER_IMAGE	**Field Type** string **Description** The default banner image for user profiles. Use a JPG, PNG, or GIF that's 1800x360 pixels and smaller than 5 MB. Users can change their profile banner image. Must refer to an asset file that already exists within the org.
USER_IMAGE	**Field Type** string **Description** The default avatar image for user profiles. Use a JPG, PNG, or GIF that's 200x200 pixels and smaller than 5 MB. Users can change their profile avatar image. Must refer to an asset file that already exists within the org.

Declarative Metadata Sample Definition

The following is an example of a BrandingSet component.

```
<?xml version="1.0" encoding="UTF-8"?>
<BrandingSet xmlns="http://soap.sforce.com/2006/04/metadata">
    <brandingSetProperty>
        <propertyName>TextTransformStyle</propertyName>
        <propertyValue>uppercase</propertyValue>
    </brandingSetProperty>
    <brandingSetProperty>
        <propertyName>DetailTextColor</propertyName>
        <propertyValue>#696969</propertyValue>
    </brandingSetProperty>
    <brandingSetProperty>
        <propertyName>BorderColor</propertyName>
        <propertyValue>#D4D4D4</propertyValue>
    </brandingSetProperty>
    <brandingSetProperty>
        <propertyName>HeaderImage</propertyName>
        <propertyValue></propertyValue>
    </brandingSetProperty>
    <brandingSetProperty>
```

```
    <propertyName>HeaderFonts</propertyName>
    <propertyValue>Montserrat</propertyValue>
</brandingSetProperty>
<brandingSetProperty>
    <propertyName>CardBackgroundColor</propertyName>
    <propertyValue>rgba(255, 255, 255, 0)</propertyValue>
</brandingSetProperty>
<brandingSetProperty>
    <propertyName>LoginBackgroundColor</propertyName>
    <propertyValue>#F4F4F4</propertyValue>
</brandingSetProperty>
<brandingSetProperty>
    <propertyName>ActionColor</propertyName>
    <propertyValue>#2574A9</propertyValue>
</brandingSetProperty>
<brandingSetProperty>
    <propertyName>_ActionColorTrans</propertyName>
    <propertyValue>rgba(25, 124, 190, 0.9)</propertyValue>
</brandingSetProperty>
<brandingSetProperty>
    <propertyName>CompanyLogo</propertyName>
    <propertyValue></propertyValue>
</brandingSetProperty>
<brandingSetProperty>
    <propertyName>LoginBackgroundImage</propertyName>
    <propertyValue>../../../../sfsites/picasso/core/external/
        salesforceIdentity/images/background.jpg?v=1</propertyValue>
</brandingSetProperty>
<brandingSetProperty>
    <propertyName>_LinkColorDarker</propertyName>
    <propertyValue>#135F90</propertyValue>
</brandingSetProperty>
<brandingSetProperty>
    <propertyName>_ActionColorDarker</propertyName>
    <propertyValue>#135F90</propertyValue>
</brandingSetProperty>
<brandingSetProperty>
    <propertyName>_HoverColor</propertyName>
    <propertyValue>rgba(25, 124, 190, 0.1)</propertyValue>
</brandingSetProperty>
<brandingSetProperty>
    <propertyName>ErrorFontColor</propertyName>
    <propertyValue>#ff9e9e</propertyValue>
</brandingSetProperty>
<brandingSetProperty>
    <propertyName>TextColor</propertyName>
    <propertyValue>#333</propertyValue>
</brandingSetProperty>
<brandingSetProperty>
    <propertyName>OverlayTextColor</propertyName>
    <propertyValue>#FFFFFF</propertyValue>
</brandingSetProperty>
<brandingSetProperty>
    <propertyName>PrimaryFont</propertyName>
```

```
        <propertyValue>Lato</propertyValue>
    </brandingSetProperty>
    <brandingSetProperty>
        <propertyName>LinkColor</propertyName>
        <propertyValue>#2574A9</propertyValue>
    </brandingSetProperty>
    <masterLabel>ex</masterLabel>
    <type>napili:branding-napili-merged</type>
</BrandingSet>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>MyBrandingSet</members>
        <name>BrandingSet</name>
    </types>
    <version>40.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

BriefcaseDefinition

Represents a briefcase definition. A briefcase makes selected records available for specific users and groups to view when they're offline in the Salesforce Field Service mobile app for iOS and Android. This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

BriefcaseDefinition components have the suffix `.briefcaseDefinition` and are stored in the `briefcaseDefinitions` folder.

Version

BriefcaseDefinition components are available in API version 50.0 and later.

Fields

Field Name	Field Type	Description
`briefcaseRules`	BriefcaseRule[]	A list of rules that specify which records are included in the briefcase.

Field Name	Field Type	Description
`description`	string	Description of the briefcase.
`isActive`	boolean	Required. Indicates whether the briefcase is active by default (`true`) or inactive (`false`). Activate a briefcase to make the selected records available to assignees.
`masterLabel`	string	Required. Label for the briefcase name that appears in the Salesforce user interface.
`type`	BriefcaseType[]	Reserved for future use.

BriefcaseRule

Represents a rule that specifies records to be included in the BriefcaseDefinition.

Field Name	Field Type	Description
`briefcaseRuleFilters`	BriefcaseRuleFilter[]	A list of filters on a rule.
`filterLogic`	string	The filter logic for record selection, for example, `1 AND 2` where 1 and 2 correspond to filter 1 and filter 2. Filter logic operators include `AND` and `OR`.
`isAscendingOrder`	boolean	Indicates whether the records should be sorted in ascending order (`true`) or descending order (`false`).
`isRelatedFilesRule`	boolean	Indicates whether the briefcase rule is part of a hierarchical set of rules that configure the offline priming of file attachments. Available only for the Offline App (Salesforce Mobile App Plus). To configure the offline priming of file attachments, create a set of four hierarchical briefcase rules: • A rule with `targetEntity` set to the object with the file attachments at the first level • A rule with `targetEntity` set to `ContentDocumentLink` at the second level • A rule with `targetEntity` set to `ContentDocument` at the third level • A rule with `targetEntity` set to `ContentVersion` at the fourth level See the Declarative Metadata Sample Definition section for an example briefcase definition that configures the offline priming of file attachments. The `ContentDocumentLink`, `ContentDocument`, and `ContentVersion` rules must all have `isRelatedFilesRule` set to `true`. To delete a briefcase configuration for file attachments, you must delete the `ContentDocumentLink` rule and all of its nested rules. You can't delete a single rule within the hierarchy of `ContentDocumentLink`, `ContentDocument`, and `ContentVersion` rules.

Field Name	Field Type	Description
		When `isRelatedFilesRule` is set to true, you must use the `recordLimit` field to limit the number of file attachments returned by a briefcase rule. Apply the same `recordLimit` value across the `ContentDocumentLink`, `ContentDocument`, and `ContentVersion` rules. You can optionally filter file attachments by file size and file type through Briefcase Builder in Setup. After you set a value for `isRelatedFilesRule`, you can no longer modify the field. The value that you set persists for the life of the rule.
`orderBy`	string	The field to order the records by, which determines how the records can be sorted. For example, Account Name or Created By.
`queryScope`	`FilterScope` (enumeration of type string)	A group of records to restrict the scope of this rule. Valid values include: • `Everything` • `AssignedToMe` • `Mine` The `AssignedToMe` scope is supported for the ServiceAppointment object only.
`recordLimit`	int	The maximum number of records for an object on the briefcase rule. The maximum is 50,000 records that meet the criteria. However, the records returned by one briefcase rule must fit within the maximum limit of 50,000 records across active briefcases. If there are more records that match the criteria than the record limit allows, the `orderBy` field determines which records are returned.
`relatedRules`	BriefcaseRule[]	A list of rules that are related to the current rule.
`relationshipField`	string	Required for `relatedRules`. Defines the Salesforce object field that relates the `relatedRules` field to another `relatedRules` field or the `briefcaseRules` field on the BriefcaseDefinition metadata type that it's nested in. For example, an Account object rule can be related to a Contact object rule using the Account ID object field. In this example, the value for the related rule's `relationshipField` is `AccountID`.
`relationshipType`	~~BriefcaseRuleRelationshipType~~ (enumeration of type string)	Required for `relatedRules`. Defines the relationship between the `relatedRules` field and another `relatedRules` field or the `briefcaseRules` field on the BriefcaseDefinition metadata type that it's nested in. Valid values include: • `ParentToChild` • `ChildToParent`
`targetEntity`	string	Required. The API name of the standard object, custom object, or custom metadata type that the briefcase rule selects records from. If the `targetEntity` is a custom metadata type, the briefcase rule can't include any other fields. You can add only one briefcase rule for the same custom metadata type in a briefcase. Custom metadata types are supported

Field Name	Field Type	Description
		as the `targetEntity` for top-level rules only–you can't create a related rule with `targetEntity` as a custom metadata type.

BriefcaseRuleFilter

Specifies filter criteria for a BriefcaseRule.

Field Name	Field Type	Description
`filterOperator`	~~BriefcaseFilterOperator~~ (enumeration of type string)	Required. The comparison operator for this rule filter. Capitalization matters with date filter operators. Be sure to specify date literals in uppercase. Some valid date literals include TODAY, YESTERDAY and TOMORROW. Valid values include: • `d`—Ends with • `e`—Equals • `g`—Greater than • `h`—Greater than or equal • `l`—Less than • `m`—Less than or equal • `n`—Not equals. This value is applicable only when `filterValue` is empty. • `s`—Starts with
`filterSeqNumber`	int	Required. The filter number. When you apply multiple filters, the filters are numbered sequentially, 1, 2, 3, and so on.
`filterValue`	string	The value that the field and criteria evaluate. For example, `true` or `false` for a boolean field whose criteria or filter operator is Equals. Be sure to specify date literals in uppercase. Some valid date literals include TODAY, YESTERDAY and TOMORROW. For `targetEntityField` values that accept a user ID, such as `OwnerId` or `CreatedById`, enter `$User.Id` to pass the ID of the user making the request. To evaluate `targetEntityField` by whether the field is empty or not empty, leave `filterValue` blank and set `filterOperator` to `e` (equals) or `n` (not equals).
`targetEntityField`	string	Required. The API name of the field to filter by. This field is from the `targetEntity` on BriefcaseRule. Compound fields aren't supported. Fields encrypted with deterministic encryption can be used in filters with equals and not equals operators.

Declarative Metadata Sample Definition

The following is an example of a BriefcaseDefinition component for account records.

The following is an example definition of a briefcase definition. If you include a rule filter, you must include a filter logic.

```
<?xml version="1.0" encoding="UTF-8"?>
<BriefcaseDefinition xmlns="http://soap.sforce.com/2006/04/metadata">
    <briefcaseRules>
        <briefcaseRuleFilters>
           <filterOperator>g</filterOperator>
           <filterSeqNumber>1</filterSeqNumber>
           <filterValue>50000.00</filterValue>
           <targetEntityField>AnnualRevenue</targetEntityField>
        </briefcaseRuleFilters>
        <briefcaseRuleFilters>
            <filterOperator>l</filterOperator>
            <filterSeqNumber>2</filterSeqNumber>
            <filterValue>50</filterValue>
            <targetEntityField>NumberOfEmployees</targetEntityField>
        </briefcaseRuleFilters>
        <filterLogic>1 AND 2</filterLogic>
        <isAscendingOrder>false</isAscendingOrder>
        <orderBy>NumberOfEmployees</orderBy>
        <queryScope>Everything</queryScope>
        <recordLimit>1000</recordLimit>
        <targetEntity>Account</targetEntity>
    </briefcaseRules>
    <description>Account Briefcase</description>
    <isActive>true</isActive>
    <masterLabel>Account With Standard Fields</masterLabel>
</BriefcaseDefinition>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>AccountWithCustomFields</members>
        <name>BriefcaseDefinition</name>
    </types>
    <version>49.0</version>
</Package>
```

This example briefcase definition configures the offline priming of file attachments for the WorkOrder object. Files Priming is available only for the Offline App (Salesforce Mobile App Plus).

```
<?xml version="1.0" encoding="UTF-8"?>
<BriefcaseDefinition xmlns="http://soap.sforce.com/2006/04/metadata"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <briefcaseRules>
       <isAscendingOrder>false</isAscendingOrder>
       <orderBy>SystemModstamp</orderBy>
       <queryScope>Everything</queryScope>
       <recordLimit>25</recordLimit>
       <targetEntity>WorkOrder</targetEntity>
```

```
        <relatedRules>
            <targetEntity>ContentDocumentLink</targetEntity>
            <relationshipField>LinkedEntityId</relationshipField>
            <relationshipType>ParentToChild</relationshipType>
            <isAscendingOrder>false</isAscendingOrder>
            <isRelatedFilesRule>true</isRelatedFilesRule>
            <queryScope>Everything</queryScope>
            <recordLimit>2</recordLimit>
            <relatedRules>
                <targetEntity>ContentDocument</targetEntity>
                <relationshipField>ContentDocumentId</relationshipField>
                <relationshipType>ChildToParent</relationshipType>
                <isAscendingOrder>false</isAscendingOrder>
                <isRelatedFilesRule>true</isRelatedFilesRule>
                <queryScope>Everything</queryScope>
                <recordLimit>2</recordLimit>
                <relatedRules>
                    <targetEntity>ContentVersion</targetEntity>
                    <relationshipField>ContentDocumentId</relationshipField>
                    <relationshipType>ParentToChild</relationshipType>
                    <isAscendingOrder>false</isAscendingOrder>
                    <isRelatedFilesRule>true</isRelatedFilesRule>
                    <queryScope>Everything</queryScope>
                    <recordLimit>2</recordLimit>
                </relatedRules>
            </relatedRules>
        </relatedRules>
    </briefcaseRules>
    <description xsi:nil="true"/>
    <isActive>true</isActive>
    <masterLabel>WorkOrder with Related Files</masterLabel>
</BriefcaseDefinition>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

Usage

Briefcase objects are available in orgs that have Briefcase Builder and Field Service enabled.

BusinessProcessGroup

Represents the surveys used to track customers' experiences across different stages in their lifecycle. This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

BusinessProcessGroup components have the suffix `.businessProcessGroup` and are stored in the `businessProcessGroups` folder.

Version

BusinessProcessGroup components are available in API version 49.0 and later.

Special Access Rules

This metadata type is available in orgs with Surveys enabled with the Customer Lifecycle Designer license.

Fields

Field Name	Field Type	Description
`businessProcessDefinitions`	BusinessProcessDefinition on page 480[]	A list that defines stages in a customer lifecycle map.
`customerSatisfactionMetric`	SurveyQuestionType (enumeration of type string)	Required. Types of questions that can be associated with stages in a customer lifecycle map. Valid values are: • `Attachment` • `Boolean` • `CSAT` • `Currency` • `Date` • `DateTime` • `FreeText` • `Image` • `NPS` • `Matrix` • `MultiChoice` • `MultiSelectPicklist` • `NPS` • `Number` • `Picklist` • `Rating` • `ShortText` • `Slider` • `StackRank` • `Toggle`

Field Name	Field Type	Description
description	string	A description of the customer lifecycle map.
masterLabel	string	Required. The name of the customer lifecycle map.

BusinessProcessDefinition

Field Name	Field Type	Description
businessProcessFeedbacks	BusinessProcessFeedback on page 480[]	A list of stages in a customer lifecycle map.
description	string	A description of a stage in the customer lifecycle map.
developerName	string	Required. The API name of a stage in the customer lifecycle map. Only users with View DeveloperName OR View Setup and Configuration permission can view, group, sort, and filter this field.
masterLabel	string	Required. The name of a stage in the customer lifecycle map.
sequenceNumber	int	Required. The position of a stage in the customer lifecycle map.

BusinessProcessFeedback

Field Name	Field Type	Description
actionName	string	Required. The name of the survey used to collect feedback
actionParam	string	Required. The name of the survey question used to collect feedback.
actionType	FeedbackCollectType (enumeration of type string)	Required. The mode of feedback collection. Valid values are: • PHONE_CALL • SURVEY

Declarative Metadata Sample Definition

The following is an example of a BusinessProcessGroup component.

```
<?xml version="1.0" encoding="UTF-8"?>
<BusinessProcessGroup xmlns="http://soap.sforce.com/2006/04/metadata">
    <businessProcessDefinitions>
        <developerName>Customer_Onboarding</developerName>
        <masterLabel>Customer Onboarding</masterLabel>
        <description>A stage in a customer's lifecycle.</description>
        <sequenceNumber>0</sequenceNumber>
        <businessProcessFeedbacks>
            <actionType>Survey</actionType>
            <actionName>New Customer CSAT</actionName>
```

```
            <actionParam>How would you rate our service?</actionParam>
        </businessProcessFeedbacks>
    </businessProcessDefinitions>
    <customerSatisfactionMetric>NPS</customerSatisfactionMetric>
    <masterLabel>Customer Lifecycle</masterLabel>
    <description>This map tracks the feedback provided by customers' at different stages
during their lifecycle.</description>
</BusinessProcessGroup>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>policyholder</members>
        <name>BusinessProcessGroup</name>
    </types>
    <version>49.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

CallCenter

Represents the Call Center definition used to integrate Salesforce with a third-party computer-telephony integration (CTI) system, a partner telephony system, or partner Contact Center as a Service (CCaaS) system.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

CallCenter components have the suffix `.callCenter` and are stored in the `callCenters` folder.

Version

CallCenter components are available in API version 27.0 and later.

Special Access Rules

There are no additional access requirements that are specific to this type.

Fields

Field Name	Description
`adapterUrl`	**Field Type** string **Description** Optional field. A URL that points to an adapter.
`contactCenterChannels`	**Field Type** ContactCenterChannel[] **Description** Relates Bring Your Own Channel for Contact Center as a Service (CCaaS) messaging channels to a CallCenter and represents the routing details for a voicemail configuration.
`displayName`	**Field Type** string **Description** Required. The display name of this call center.
`displayNameLabel`	**Field Type** string **Description** Required. The label of the `displayName` field in Call Center setup page.
`internalNameLabel`	**Field Type** string **Description** Required. The label of the `internalName` field in Call Center setup page.
`sections`	**Field Type** CallCenterSection[] **Description** Custom setup items defined for this call center.
`version`	**Field Type** string **Description** The version of this call center.

CallCenterSection

Field Name	Description
`items`	**Field Type** CallCenterItem[] **Description** Contains the label, name, and value that describe the sections.
`label`	**Field Type** string **Description** Required. The label of the section.
`name`	**Field Type** string **Description** Required. The name of the section.

CallCenterItem

Field Name	Description
`label`	**Field Type** string **Description** Required. The label of the custom setup item.
`name`	**Field Type** string **Description** Required. The name of the custom setup item.
`value`	**Field Type** string **Description** Required.

Field Name	Description
	The value of the custom setup item.

ContactCenterChannel

Represents a junction subtype that relates a Bring Your Own Channel for Contact Center as a Service (CCaaS) messaging channel to a CallCenter type for Bring Your Own Channel for CCaaS. This subtype also represents the routing details for a voicemail configuration. This subtype is available in API version 56.0 and later.

Field Name	Description
`channel`	**Field Type** string **Description** Required. For Bring Your Own Channel for CCaaS, this field represents the unique ID of the Bring Your Own Channel messaging channel (MessagingChannel) that's associated with the contact center (CallCenterId). Available in API version 60.0 and later.
`contactCenter`	**Field Type** string **Description** Required. For Bring Your Own Channel for CCaaS, this field represents the unique ID of the contact center (CallCenterId) that's associated with the Bring Your Own Channel messaging channel (MessagingChannel). Available in API version 60.0 and later.
`voiceMailFallbackQueue`	**Field Type** string **Description** If voicemail routing is configured for the contact center, this field represents the unique ID of the fallback queue to use if voicemail routing fails. Don't change the value in this field. Instead, configure voicemail routing in Lightning Experience.
`voiceMailHandler`	**Field Type** string **Description** If voicemail routing is configured for the contact center, this field represents the unique ID of the flow used to route voicemails. Don't change the value in this field. Instead, configure voicemail routing in Lightning Experience.

Declarative Metadata Sample Definition

The following is an example of a CallCenter component:

```
<?xml version="1.0" encoding="UTF-8"?>
<CallCenter xmlns="http://soap.sforce.com/2006/04/metadata">
    <adapterUrl>http://localhost:11000</adapterUrl>
    <displayName>Demo Call Center Adapter</displayName>
    <displayNameLabel>Display Name</displayNameLabel>
    <internalNameLabel>Internal Name</internalNameLabel>
    <sections>
        <items>
            <label>Description</label>
            <name>reqDescription</name>
            <value>Demo Call Center Adapter</value>
        </items>
        <items>
            <label>CTI Connector ProgId</label>
            <name>reqProgId</name>
            <value>DemoAdapter.DemoAdapter.1</value>
        </items>
        <items>
            <label>Version</label>
            <name>reqVersion</name>
            <value>3.0</value>
        </items>
        <items>
            <label>CTI Adapter URL</label>
            <name>reqAdapterUrl</name>
            <value>http://localhost:11000</value>
        </items>
        <label>General Information</label>
        <name>reqGeneralInfo</name>
    </sections>
    <sections>
        <items>
            <label>Outside Prefix</label>
            <name>reqOutsidePrefix</name>
            <value>1</value>
        </items>
        <items>
            <label>Long Distance Prefix</label>
            <name>reqLongDistPrefix</name>
            <value>1</value>
        </items>
        <items>
            <label>International Prefix</label>
            <name>reqInternationalPrefix</name>
            <value>01</value>
        </items>
        <label>Dialing Options</label>
        <name>reqDialingOptions</name>
    </sections>
    <version>4</version>
</CallCenter>
```

For information about the CallCenter definition file, see Call Center Definition Files.

CallCenterRoutingMap

Represents the mapping between a user or queue in a Salesforce org to a user or queue in an external system's call center.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

CallCenterRoutingMap components have the suffix `.callCenterRoutingMap` and are stored in the `callCenterRoutingMaps` folder.

Version

CallCenterRoutingMap components are available in API version 52.0 and later.

Special Access Rules

This type requires Contact Center Admin, Contact Center Admin (Partner Telephony), Contact Center Supervisor, or Manage Call Centers permission.

Fields

Field Name	Description
`callCenter`	**Field Type** string **Description** Required. Reference to a call center.
`developerName`	**Field Type** string **Description** Required. The developer name is a combination of the Salesforce user ID or queue name, and the `callCenter` value, with an underscore between these two values. • `[SALESFORCE_USER_ID]_[CALL_CENTER]`

Field Name	Description
	• [SALESFORCE_QUEUE_NAME]_[CALL_CENTER]
externalId	**Field Type** string **Description** Required. Unique identifier for the external system's user or queue.
masterLabel	**Field Type** string **Description** Required. The master label of the CallCenterRoutingMap.
quickConnect	**Field Type** string **Description** The Amazon Connect QuickConnectId ARN used to determine agent availability for Omni-Channel call transfers. Available in API version 56.0 and later.
referenceRecord	**Field Type** string **Description** Required. Lookup field to a Salesforce user or queue.

Declarative Metadata Sample Definition

The following is an example of a CallCenterRoutingMap component.

```
<?xml version="1.0" encoding="UTF-8"?>
<CallCenterRoutingMap xmlns="http://soap.sforce.com/2006/04/metadata">
    <callCenter>ExampleCallCenter</callCenter>
<developerName>User_001ABC00000FjYIIA0_04vZ6000000Cagl</developerName>
<externalId>arn:aws:connect:ap-northeast-1:484907484500:instance/examplestring-9c18-4aa5-b5fe-cd6f34d99a0/agent/a69f7afe-5b04-4aa8-b5ee-108a84d0f504</externalId>

    <masterLabel>001ABC00000FjYIIA0</masterLabel>
    <referenceRecord>example.d2b87b8182fa@salesforce.com</referenceRecord>
</CallCenterRoutingMap>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
```

```
    <types>
        <members>User_001ABC00000FjYIIA0_04vZ6000000Cagl</members>
        <name>CallCenterRoutingMap</name>
    </types>
    <version>64.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

CallCoachingMediaProvider

Represents the CallCoachingMediaProvider configuration. Use CallCoachingMediaProvider to configure which providers of voice recordings that Einstein Conversation Insights can use. For example, Sales Dialer can provide voice recordings. Einstein Conversation Insights then stores and analyzes call recordings to surface insights and trends in customer conversations.This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

CallCoachingMediaProvider components have the suffix `.callCoachingMediaProvider` and are stored in the `CallCoachingMediaProvider` folder.

Version

CallCoachingMediaProvider components are available in API version 49.0 and later.

Special Access Rules

You must be a Sales Engagement customer to access this metadata type.

Fields

Field Name	Field Type	Description
`isActive`	boolean	Indicates whether the media provider can upload voice recordings (`true`) or not (`false`). Default value is `false`.
`providerDescription`	string	Description of the media provider.
`providerName`	string	Name of the media provider.

Declarative Metadata Sample Definition

The following is an example of a CallCoachingMediaProvider component.

```
<?xml version="1.0" encoding="UTF-8"?>
<CallCoachingMediaProvider xmlns="http://soap.sforce.com/2006/04/metadata">
    <isActive>true</isActive>
    <providerDescription>Salesforce telephony provider</providerDescription>
    <providerName>Dialer</providerName>
</CallCoachingMediaProvider>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>CallCoachingMediaProvider</name>
    </types>
    <version>49.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

CampaignInfluenceModel

Represents a campaign influence model used by Customizable Campaign Influence. You can't configure Customizable Campaign Influence via the Metadata API, but you can add a campaign influence model.

Note: This information applies only to Customizable Campaign Influence and not to Campaign Influence 1.0 .

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

CampaignInfluenceModel values are stored in the `campaignInfluenceModels` directory of the corresponding package directory. The file name matches the model name, and the extension is `.campaignInfluenceModel`.

Version

CampaignInfluenceModel components are available in API version 38.0 and later.

Fields

Field Name	Field Type	Description
`isActive`	boolean	Indicates whether the model is active. Active models can generate campaign influence records. Deactivating a model deletes its campaign influence records. Custom models are always active and this field is ignored. This field is available beginning with API version 40.0.
`isDefaultModel`	boolean	Required. Indicates if the model is the default model or not. Only campaign influence records associated with the default model appear on campaigns and opportunities. You can only have one default model at a time. A model must be active to become the default model. Activating or deactivating custom models does not automatically generate or delete campaign influence records.
`isModelLocked`	boolean	Required. Indicates if the model is locked or not. Campaign Influence records for locked models can be manipulated only via the API.
`modelDescription`	string	A description of the influence model.
`name`	string	Required. A unique name for the model.
`recordPreference`	picklist	The value of this field determines when to create campaign influence records. • `AllRecords`: Creates records regardless of the revenue attribution percentage. • `RecordsWithAttribution`: Creates records only when the revenue attribution is greater than 0%. This field is available In API version 41.0 and later.

Declarative Metadata Sample Definition

The following is an example of a CampaignInfluenceModel component that represents the default Salesforce campaign influence attribution model. The default `isDefaultModel` value of `true` can be changed if another model is created and set as the default model. The `isModelLocked` value of `true` means that Campaign Influence records for this model can be seen in the UI, but not created, updated, or deleted.

```
<?xml version="1.0" encoding="UTF-8"?>
<CampaignInfluenceModel xmlns="http://soap.sforce.com/2006/04/metadata">
    <isActive>true</isActive>
    <isDefaultModel>true</isDefaultModel>
   <isModelLocked>true</isModelLocked>    <recordPreference>AllRecords</recordPreference>

    <modelDescription>Primary Campaign gets 100% of the revenue share</modelDescription>
    <name>Salesforce Model</name>
</CampaignInfluenceModel>
```

The following is an example of a CampaignInfluenceModel component that creates an influence model called Last Touch, which will not be the default model.

```
<?xml version="1.0" encoding="UTF-8"?>
<CampaignInfluenceModel xmlns="http://soap.sforce.com/2006/04/metadata">
    <isActive>true</isActive>
    <isDefaultModel>false</isDefaultModel>
    <isModelLocked>true</isModelLocked>
    <modelDescription>This model gives 100% influence attribution to the last campaign
that touched the contact.</modelDescription>
    <name>Last Touch</name>
    <recordPreference>RecordsWithAttribution</recordPreference>
</CampaignInfluenceModel>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

CaseSubjectParticle

Represents the Social Business Rules custom format for the **Case Subject** field on cases created from inbound social posts.

File Suffix and Directory Location

CaseSubjectParticle components have the suffix `.CaseSubjectParticle` and are stored in the `CaseSubjectParticles` folder.

Version

CaseSubjectParticle is available in API version 41.0 and later.

Fields

Field Name	Field Type	Description
`index`	int	Required. The order in which the custom **Case Subject** is generated, meaning if the social network is 0 and the social message is 1, then the subject generates as `Twitter | Tweet`.
`textField`	string	Specifies inbound social content added to **Case Subject** in case records.

Field Name	Field Type	Description
`type`	CaseSubjectParticleType (enumeration of type string)	Required. Specifies the custom **Case Subject** format from which inbound social content appears in case records. Valid values are: • `ProvidedString` • `Source` • `MessageType` • `SocialHandle` • `SocialNetwork` • `Sentiment` • `RealName` • `Content` • `PipeSeparator` • `ColonSeparator` • `HyphenSeparator`

Declarative Metadata Sample Definition

This is a sample of a `.CaseSubjectParticle` file.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns=http://soap.sforce.com/2006/04/metadata"">
    <types>
        <members>*</members>
        <name>CaseSubjectParticle</name>
    </types>
    <version>41.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type doesn't support the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

CareBenefitVerifySettings

Represents the configuration settings for benefit verification requests.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

CareBenefitVerifySettings components have the suffix `.careBenefitVerifySettings` and are stored in the `careBenefitVerifySettings` folder.

Version

CareBenefitVerifySettings components are available in API version 52.0 and later.

Fields

Field Name	Description
`codeSetType`	**Field Type** string **Description** Specifies the code set type for the benefits verification service type codes.
`defaultNpi`	**Field Type** string **Description** Default National Provider Identifier to be used in the benefits verification request.
`generalPlanServiceTypeCode`	**Field Type** string **Description** Service type code for the plan benefits as a whole.
`isDefault`	**Field Type** boolean **Description** Indicates whether this record is the default verification service `(true)` or not `(false)`.
`masterLabel`	**Field Type** string **Description** Required. Name of the benefits verification service.
`organizationName`	**Field Type** string **Description** Specifies the organization name for the benefits verification request service.

Field Name	Description
serviceApexClass	**Field Type** string **Description** Apex class used to access the benefits verification service.
serviceNamedCredential	**Field Type** string **Description** Credential used to access the benefits verification service.
serviceTypeSourceSystem	**Field Type** string **Description** Service type code for the plan benefits as a whole.
uriPath	**Field Type** string **Description** Link to payer endpoint.

Declarative Metadata Sample Definition

This is an example of a CareBenefitVerifySettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<CareBenefitVerifySettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <generalPlanServiceTypeCode>abc</generalPlanServiceTypeCode>
    <isDefault>true</isDefault>
    <masterLabel>test</masterLabel>
    <serviceApexClass>TestApexClass</serviceApexClass>
    <serviceNamedCredential>test</serviceNamedCredential>
    <uriPath>efgh</uriPath>
    <serviceTypeSourceSystem>Lorem ipsum dolor</serviceTypeSourceSystem>
    <codeSetType>Code set</codeSetType>
    <defaultNpi>Npi info</defaultNpi>
    <organizationName>Organization name</organizationName>
</CareBenefitVerifySettings>
```

This is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>CareBenefitVerifySettings</name>
    </types>
    <types>
```

```
        <members>*</members>
        <name>ApexClass</name>
    </types>
    <types>
        <members>*</members>
        <name>NamedCredential</name>
    </types>
    <version>52.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

CareLimitType

Defines the characteristics of limits on benefit provision.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

CareLimitType components have the suffix `.careLimitType` and are stored in the `careLimitTypes` folder.

Version

CareLimitType components are available in API version 52.0 and later.

Fields

Field Name	Description
`isProtected`	**Field Type** boolean **Description** An auto-generated value that doesn't impact the behavior of the metadata type.
`limitType`	**Field Type** string

Field Name	Description
	Description Source of limit on benefit provision, such as a co-insurance requirement.
`masterLabel`	**Field Type** string **Description** Required. Name of the limit type.
`metricType`	**Field Type** CareLimitTypeMetricType (enumeration of type string) **Description** Metric to be used for calculating and displaying the benefit limit, such as number of visits, amount spent, or percentage of allowed expenditure. Valid values are: • `Amount` • `Money` • `Percentage` • `Text`

Declarative Metadata Sample Definition

This is an example of a CareLimitType component.

```
<?xml version="1.0" encoding="UTF-8"?>
<CareLimitType xmlns="http://soap.sforce.com/2006/04/metadata">
    <limitType>test</limitType>
    <masterLabel>test</masterLabel>
    <metricType>Money</metricType>
    <isProtected>false</isProtected>
</CareLimitType>
```

This is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>CareLimitType</name>
    </types>
    <version>52.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

CareSystemFieldMapping

Represents a mapping from source system fields to Salesforce objects and fields. This type extends the Metadata metadata type and inherits its `fullName` field.

[other]: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

CareSystemFieldMapping components have the suffix `.careSystemFieldMapping` and are stored in the `careSystemFieldMappings` folder.

Version

CareSystemFieldMapping components are available in API version 49.0 and later.

Special Access Rules

To use this metadata type, your Salesforce org must have the Health Cloud or Life Sciences Cloud license and the user must have the Health Cloud Foundation (for Health Cloud) or Health Cloud Starter (for Life Sciences Cloud) permission set.

Fields

Field Name	Field Type	Description
`externalIdField`	string	The ID of the field in the external system.
`isActive`	boolean	Indicates whether this field mapping is active (`true`) or not (`false`). The default value is False.
`isProtected`	boolean	An auto-generated value that doesn't currently impact the behavior of the metadata type.
`masterLabel`	string	Required. The name of the care system field mapping.
`role`	SourceSystemFieldRole (enumeration of type string)	Required. The role the field represents. Valid values are: • `Patient`—When the `role` field is set to `Patient`, the Enrollment API uses the value of `externalIdField` as the patient ID. This role can be used when `targetObject` is set to `Account`. • `RemoteMonitoringDevice`—Indicates which `externalIdField` on the Asset object maps to the `Device`

Field Name	Field Type	Description
		field in the CareObservation object. This role can be used when targetObject is set to Asset. • `RemoteMonitoringPatient`—Indicates which `externalIdField` on the Account object maps to the `ObservedSubject` field in the Care Observation object. This role is used when targetObject is set to Account. • `ServiceProvider`—The Enrollment API uses the value of `externalIdField` as the provider ID. This role is used when `targetObject` is set to `Account`. • `NotApplicable`—This role is used when `targetObject` is set to `CareProgram` or `Product`, which means that there is no applicable role.
`sourceSystem`	string	The system where the record originated.
`targetObject`	string	The name of the Salesforce object to which the external system field is mapped.

Declarative Metadata Sample Definition

The following is an example of a CareSystemFieldMapping component.

```
<?xml version="1.0" encoding="UTF-8"?>
<CareSystemFieldMapping xmlns="http://soap.sforce.com/2006/04/metadata">
    <externalIdField>AccountNumber</externalIdField>
    <isActive>true</isActive>
    <isProtected>false</isProtected>
    <masterLabel>Map1</masterLabel>
    <role>Patient</role>
    <sourceSystem>Epic</sourceSystem>
    <targetObject>Account</targetObject>
</CareSystemFieldMapping>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>patient</members>
        <name>CareSystemFieldMapping</name>
    </types>
    <version>49.0</version>
</Package>
```

CareProviderSearchConfig

Represents the information about the fields that appear in care provider search results.This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

CareProviderSearchConfig components have the suffix `.careProviderSearchConfig` and are stored in the `careProviderSearchConfigs` folder.

Version

CareProviderSearchConfig components are available in API version 48.0 and later.

Fields

Field Name	Field Type	Description
`isActive`	boolean	Indicates whether this configuration is active (`true`) or not (`false`).
`isProtected`	boolean	An auto-generated value that doesn't currently impact the behavior of the metadata type.
`mappedObject`	ProviderSearch ObjectMapping (enumeration of type string)	Required. Indicates mapped objects. Possible values are; • HealthCarePractitionerFacility • HealthCareProvider
`masterLabel`	string	Required. Name of the care provider.
`sourceField`	string	API name of the field that is copied to the target object.
`targetField`	string	API name of the field to copy the data to.

Declarative Metadata Sample Definition

The following is an example of a CareProviderSearchConfig component.

```
<?xml version="1.0" encoding="UTF-8"?>
<CareProviderSearchConfig xmlns="http://soap.sforce.com/2006/04/metadata">
    <sourceField>Test1__c</sourceField>
    <targetField>Test1__c</targetField>
    <mappedObject>HealthcareProvider</mappedObject>
    <isProtected>false</isProtected>
    <isActive>true</isActive>
```

```
    <masterLabel>testlabel</masterLabel>
</CareProviderSearchConfig>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>HealthcareProvider.Test1__c</members>
        <name>CustomField</name>
    </types>
    <types>
        <members>CareProviderSearchableField.Test1__c</members>
        <name>CustomField</name>
    </types>
    <types>
        <members>Test</members>
        <name>CareProviderSearchConfig</name>
    </types>
    <version>48.0</version>
</Package>
```

CareRequestConfiguration

Represents the details for a record type such as service request, drug request, or admission request. One or more record types can be associated with a care request.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

CareRequestConfiguration components have the suffix `.careRequestConfiguration` and are stored in the `careRequestConfigurations` folder.

Version

CareRequestConfiguration components are available in API version 44.0 and later.

Fields

Field Name	Description
`careRequestRecordType`	**Field Type** string **Description** Required. The record type for the care request.
`careRequestRecords`	**Field Type** CareRequestRecords[] **Description** The list of objects you can select to configure the care request.
`careRequestType`	**Field Type** string **Description** Required. The type of care request. For example, an appeal, a service request, or an admission.
`isActive`	**Field Type** boolean **Description** Indicates whether the care request is active (`true`) or not (`false`).
	Field Type boolean **Description** Indicates whether the record type of the care request is default (`true`) or not (`false`).
	Field Type string **Description** Required. A user-friendly name for CareRequestConfiguration, which is defined when the CareRequestConfiguration is created.

CareRequestRecords

Displays a list of objects to customize the care request.

Field Name	Description
careRequestRecord	**Field Type** string **Description** Required. The object selected to configure the care request.

Declarative Metadata Sample Definition

This is an example of a CareRequestConfiguration component.

```
<?xml version="1.0" encoding="UTF-8"?>
<CareRequestConfiguration xmlns="http://soap.sforce.com/2006/04/metadata">
    <careRequestRecordType>DrugRequest</careRequestRecordType>
    <careRequestRecords>
        <careRequestRecord>CareRequestItem</careRequestRecord>
    </careRequestRecords>
    <careRequestRecords>
        <careRequestRecord>CareRequestDrug</careRequestRecord>
    </careRequestRecords>
    <careRequestType>Drug Request</careRequestType>
    <isActive>false</isActive>
    <isDefaultRecordType>false</isDefaultRecordType>
    <masterLabel>DrugRequest</masterLabel>
</CareRequestConfiguration>
```

This is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>Case.DrugRequest</members>
        <name>BusinessProcess</name>
    </types>
    <types>
        <members>*</members>
        <name>CareRequestConfiguration</name>
    </types>
    <types>
        <members>CareRequest.DrugRequest</members>
        <members>CareRequestDrug.DrugRequest</members>
        <members>CareRequestItem.DrugRequest</members>
        <members>Case.DrugRequest</members>
        <name>RecordType</name>
    </types>
    <version>44.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

Certificate

Represents a certificate used for digital signatures that verify that requests are coming from your org. Certificates are used for either authenticated single sign-on with an external website, or when using your org as an identity provider. This type extends the Metadata With Content metadata type and inherits its `content` and `fullName` fields.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

Certificate components have the suffix `.crt` and are stored in the `certs` folder.

Version

Certificate components are available in API version 36.0 and later.

Fields

Field Name	Field Type	Description
`caSigned`	boolean	Required. Indicates whether this certificate is signed by the issuer (true) or not (false).
`encryptedWithPlatformEncryption`	boolean	Indicates whether this certificate is encrypted with Platform Encryption.
`expirationDate`	dateTime	The date that this certificate expires and is no longer usable. For self-signed certificates, if `keySize` is 2048 bits, the expiration date is automatically 1 year after you create the certificate. If `keySize` is 4096 bits, the expiration date is automatically 2 years after you create the certificate. For CA-signed certificates, `expirationDate` is automatically updated to the signed certificate's expiration date when a signed certificate chain is uploaded. The date format is YYYY-MM-DD.
`keySize`	int	Certificate keys can be either 2048 bits or 4096 bits. A certificate with 4096-bit keys lasts 2 years, and a certificate with 2048-bit keys lasts 1 year. Certificates with 2048-bit keys are faster than certificates with 4096-bit keys. If `keySize` isn't specified when you create a certificate, the key size defaults to 2048 bits.
`masterLabel`	string	Required. A user-friendly name for the certificate that appears in the Salesforce user interface, such as in Certificate and Key Management. Limit: 64 characters.

Field Name	Field Type	Description
`privateKeyExportable`	boolean	Indicates whether this certificate's private key is exportable. If `privateKeyExportable` isn't specified when you create a certificate, its default value is `true`.

Usage

The Metadata API can be used to create a self-signed or a CA-signed certificate. The .crt file's contents are the certificate chain, which can be updated when you renew or update the intermediate certificate chain of a CA-signed certificate. After creating a CA-signed certificate, the .crt file contains a certificate signing request (CSR). For details, see About Salesforce Certificates and Keys in Salesforce Help.

To copy an existing certificate's X.509 parameter data to a new certificate, upload the existing certificate. You can also use this procedure to renew a certificate. A new private+public key pair is created with a new certificate. Salesforce doesn't allow the import or export of the private key via the API.

Using the Metadata API, you can download a CSR. After it's CA-signed, you can upload it back to Salesforce.

After the signed certificate chain is uploaded via the Metadata API, the CSR of that certificate can't be downloaded via the API anymore. The content of the `.crt` file is the signed certificate chain. However, the CSR can still be downloaded via the UI.

- Downloading a CSR—The CSR is downloadable after a CA-signed cert is created. If a signed certificate hasn't been uploaded to that certificate, the content of the downloaded .crt file is the CSR.
- Uploading a CA-Signed Certificate—To upload the signed certificate chain back to Salesforce, save the signed certificate chain as the content of the .crt file and update it via the Metadata API.

Declarative Metadata Sample Definition

The following is an example of a Certificate component.

```
<?xml version="1.0" encoding="UTF-8"?>
<Certificate xmlns="http://soap.sforce.com/2006/04/metadata">
    <caSigned>true</caSigned>
    <encryptedWithPlatformEncryption>true</encryptedWithPlatformEncryption>
    <expirationDate>2017-03-19</expirationDate>
    <keySize>4096</keySize>
    <masterLabel>My Certificate Name</masterLabel>
    <privateKeyExportable>true</privateKeyExportable>
</Certificate>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ChatterExtension

Represents the metadata used to describe a Rich Publisher App that's integrated with the Chatter publisher.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Retrieving ChatterExtension

Using an API tool, you can get extension information from `package.xml` using this code.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
 <types>
  <members>xw1</members>
  <name>ChatterExtension</name>
 </types>
 <version>41.0</version>
</Package>
```

Use the `<members>` tag to name a specific extension (in this example, *xw1*), or use the wildcard (*) symbol to retrieve all your extensions.

Here's an example of retrieved information.

```
<?xml version="1.0" encoding="UTF-8"?>
<ChatterExtension xmlns="http://soap.sforce.com/2006/04/metadata">
  <compositionComponent>xwComp</compositionComponent>
  <description>des</description>
  <extensionName>xw1</extensionName>
  <headerText>h1</headerText>
  <hoverText>h2</hoverText>
  <icon>tiger</icon>
  <masterLabel>primary</masterLabel>
  <renderComponent>xwRend</renderComponent>
  <type>Lightning</type>
</ChatterExtension>
```

Version

ChatterExtension is a new feature in API version 41.0.

Fields

Field	Field Type	Description
`compositionComponent`	string	Required. The composition component of the Rich Publisher App that you provide. It's comprised of the `lightning:availableForChatterExtensionComposer` interface.
`description`	string	Required. The description of your custom Rich Publisher App.
`extensionName`	string	Required. The name of your extension. That is, your Rich Publisher App.

Field	Field Type	Description
`headerText`	string	The text to show in the header of your app composer. Header text is required for Lightning type extensions. This text can be localized.
`hoverText`	string	The text to show when a user mouses over your extension's icon. Mouse-over text is required for Lightning type extensions. This text can be localized.
`icon`	string	Required. The icon to show in the Chatter publisher. Use an existing file asset id from your org.
`isProtected`	boolean	An auto-generated value. It currently has no impact.
`masterLabel`	string	Required. Label for the ChatterExtension object.
`renderComponent`	string	Required. The rendering component of the Rich Publisher App that you provide. It's comprised of the `lightning:availableForChatterExtensionRenderer` interface.
`type`	ChatterExtensionType (enumeration of type string)	Required. Describes the type of the extension. Currently, the only value supported is *`Lightning`*. Included to allow for other possible types in the future.

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SEE ALSO:

Integrate Your Custom Apps into the Chatter Publisher

ChoiceList

Represents the `Choicelist` dropdown field that's used for pre-chat.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

ChoiceList components have the suffix `.ChoiceList` and are stored in the `ChoiceList` folder.

Version

ChoiceList components are available in API version 62 and later.

Special Access Rules

There are no additional access requirements that are specific to this type.

Fields

Field Name	Description
`choiceListValue`	**Field Type** ChoiceListValue[] **Description** A list of choices to display in the choice list.
`description`	**Field Type** string **Description** A description of the choice list.
`masterLabel`	**Field Type** string **Description** Required. The label for the choice list.

ChoiceListValue

Represents a choice list value in the pre-chat dropdown. ChoiceListValue is available in API version 62 or later.

Field Name	Description
`embeddedServiceCustomLabels`	**Field Type** EmbeddedServiceCustomLabel[] on page 915 **Description** Custom labels for the choicelist value.
`isDefaultValue`	**Field Type** boolean **Description** Required. Indicates whether the choicelist value should be selected by default.

Field Name	Description
order	**Field Type** int **Description** Required. The order of the choicelist value in the choicelist dropdown field.
valueName	**Field Type** string **Description** Required. The value of the choicelist.

Declarative Metadata Sample Definition

The following is an example of a Choicelist component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ChoiceList xmlns="http://soap.sforce.com/2006/04/metadata">
    <choiceListValue>
        <embeddedServiceCustomLabels>

<customLabel>EM_PreChat_ChoiceList_PrechatCustomFieldLabel_133xx0000004GG1_7741637</customLabel>

            <labelKey>EM_PreChat_ChoiceList_PrechatCustomFieldLabel</labelKey>
            <feature>EmbeddedMessaging</feature>
        </embeddedServiceCustomLabels>
        <isDefaultValue>true</isDefaultValue>
        <order>0</order>
        <valueName>Pizza</valueName>
    </choiceListValue>
    <choiceListValue>
        <embeddedServiceCustomLabels>

<customLabel>EM_PreChat_ChoiceList_PrechatCustomFieldLabel_133xx0000004GG2_5523047</customLabel>

            <labelKey>EM_PreChat_ChoiceList_PrechatCustomFieldLabel</labelKey>
            <feature>EmbeddedMessaging</feature>
        </embeddedServiceCustomLabels>
        <isDefaultValue>false</isDefaultValue>
        <order>1</order>
        <valueName>Burger</valueName>
    </choiceListValue>
    <masterLabel>Food</masterLabel>
    <description>Food Choice List</description>
</ChoiceList>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
```

```
    <types>
        <members>*</members>
        <name>ChoiceList</name>
    </types>
    <version>62.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ClaimFinancialSettings

Represents the configuration settings for Insurance Claim Financial Services.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

ClaimFinancialSettings components have the suffix `claimFinancialSettings` and are stored in the `ClaimFinancialSettings` folder.

Version

ClaimFinancialSettings components are available in API version 57.0 and later.

Special Access Rules

To access this metadata type, you require access to either InsurancePolicyAdminAccess or InsuranceClaimMgmtAccess add-on license.

Fields

Field Name	Description
`claimCovPendingAuthStatus`	**Field Type** string **Description** Required. The status of pending financial authority for claim coverage.

Field Name	Description
claimPendingAuthorityStatus	**Field Type** string **Description** Required. The status of pending financial authority for claim.
clmCovPymtDtlPendAuthSts	**Field Type** string **Description** Required. The status of pending financial authority for claim coverage payment detail.
masterLabel	**Field Type** string **Description** Required. The unique label that identifies the claim financial settings throughout the Salesforce user interface.

Declarative Metadata Sample Definition

The following is an example of a ClaimFinancialSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ClaimFinancialSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <claimCovPendingAuthStatus>Pending Authority</claimCovPendingAuthStatus>
    <claimPendingAuthorityStatus>Pending Authority</claimPendingAuthorityStatus>
    <clmCovPymtDtlPendAuthSts>Pending Authority</clmCovPymtDtlPendAuthSts>
    <masterLabel>Claim Financial Settings</masterLabel>
</ClaimFinancialSettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?><!--
  ~ Copyright 2022 salesforce.com, inc.
  ~ All Rights Reserved
  ~ Company Confidential
  -->
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>ClaimFinancialSettings</name>
    </types>
    <version>57.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ClauseCatgConfiguration

Represents the configuration about the clause category that can be used to categorize your disclosure and compliance reports from standardized disclosure templates in a response document.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

ClauseCatgConfiguration components have the suffix `.clauseCatgConfiguration` and are stored in the `clauseCatgConfigurations` folder.

Version

ClauseCatgConfiguration components are available in API version 57.0 and later.

Special Access Rules

The ClauseManagementAddOn license is required to access this object along with user access for the Clause Designer User permission set license.

Fields

Field Name	Description
`description`	**Field Type** string **Description** The description about the clause category configuration.
`isProtected`	**Field Type** boolean **Description** An auto-generated value that doesn't impact the behavior of the metadata type. The default is `false`.
`masterLabel`	**Field Type** string

Field Name	Description
	Description Required. A user-friendly name for ClauseCatgConfiguration, which is defined when the ClauseCatgConfiguration is created.
`usageType`	**Field Type** ClmCategoryUsageType **Description** Required. The usage type of the clause category configuration. Possible values are: • `ContractClauseCategory` • `DisclosureCategory`

Declarative Metadata Sample Definition

The following is an example of a ClauseCatgConfiguration component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ClauseCatgConfiguration
 xmlns="http://soap.sforce.com/2006/04/metadata">
 <description>This is to add description for Contract Clause Category.</description>
 <usageType>ContractClauseCategory</usageType>
 <isProtected>false</isProtected>
 <masterLabel>Contract Clause Cat</masterLabel>
</ClauseCatgConfiguration>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package
 xmlns="http://soap.sforce.com/2006/04/metadata">
 <fullName>Pkg</fullName>
 <types>
  <name>ClauseCatgConfiguration</name>
 </types>
 <version>57.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

CleanDataService

Represents a data service that adds and updates data in standard objects.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

CleanDataService components have the `.cleanDataService` suffix and are stored in the `cleanDataServices` directory. The name of the component file is based on the name of the object associated with the data service. For example, the component file name `cleanDataServices/DataCloudCompanyMatch.cleanDataService` describes a data service component called DataCloudCompanyMatch that's associated with the company name in account objects.

Version

CleanDataService components are available in API version 64.0 and later.

Fields

Field Name	Field Type	Description
`cleanRules`	CleanRule[]	Required. A list of data integration rules
`description`	string	Required. A description of the data service
`masterLabel`	string	Required. Label for this data service. Although this value is displayed, it's an internal label for the data service and isn't translated.
`matchEngine`	string	Required. A key that maps to the internal data service identifier.

CleanRule

Represents information that controls how the data service adds and updates data in an org.

Field Name	Field Type	Description
`bulkEnabled`	boolean	Required. If this field is set to `true`, Salesforce applies the data integration rule to existing records whenever the rule is updated or saved.
`bypassTriggers`	boolean	Required. If this field is set to `true`, Salesforce bypasses triggers when it applies the rule; otherwise, it applies triggers after it applies the rule.
`bypassWorkflow`	boolean	Required. If this field is set to `true`, Salesforce bypasses workflow rules when it applies the data integration rule; otherwise, it applies workflow rules after it applies the rule.
`description`	string	Required. User-friendly text that describes the data integration rule.

Field Name	Field Type	Description
developerName	string	Required. This name can contain only underscores and alphanumeric characters, and must be unique in your org. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. This unique name prevents conflicts with rules from other packages that have the same masterLabel. Note: Only users with View DeveloperName OR View Setup and Configuration permission can view, group, sort, and filter this field.
fieldMappings	FieldMapping[]	Required. A list of FieldMapping entries for the rule.
masterLabel	string	Required. Label for this object. This display value is the internal label that is not translated.
matchRule	string	Required. An internal label for a matching rule in the data service that's associated with the CleanRule.
sourceSobjectType	string	Required. A virtual object in the data service that is associated with the CleanRule. Specifying a non-existent object causes an error.
status	string	Required. Status of the data integration rule. Valid values are Active and Inactive.
targetSobjectType	string	Required. A standard object that's the target of additions and updates specified by this CleanRule. Specifying an object that the data service does not support causes an error.

FieldMapping

Represents a mapping between fields in the data service and fields in an object in the org.

Field Name	Field Type	Description
developerName	string	Required. This name can contain only underscores and alphanumeric characters, and must be unique in your org. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. This unique name prevents conflicts with field mappings from other packages that have the same masterLabel. Note: Only users with View DeveloperName OR View Setup and Configuration permission can view, group, sort, and filter this field.
fieldMappingRows	FieldMappingRow[]	Required. A list of FieldMappingRow entries. Each entry represents a field in a standard object that maps to a field in the data service.
masterLabel	string	Required. Label for this object. This display value is the internal label that is not translated.
SObjectType	string	Required. The standard object associated with this FieldMapping. Specifying an object that the data service does not support causes an error.

FieldMappingRow

Represents the status of a CleanRule.

Field Name	Field Type	Description
`fieldName`	string	The display name for the field represented by the FieldMappingRow.
`fieldMappingFields`	FieldMappingField[]	Required. A list of FieldMappingField entries. Each entry is a field in a standard object that maps to a field in the data service.
`mappingOperation`	string	The comparison operation the data service applies when it compares the value of this FieldMappingRow to the mapped field in the object specified in SObjectType. The value of this field is `AutoFill`, which indicates that the data service only adds data if the object field is blank.
`SObjectType`	string	The standard object for the field mapped to the FieldMappingRow. Specifying an object that the data service does not support causes an error.

FieldMappingField

Represents a field in a standard object. A FieldMappingField maps to a FieldMappingRow entry in a data service.

Field Name	Field Type	Description
`dataServiceField`	string	Required. A field in the data service that is mapped to this field.
`dataServiceObjectName`	string	Required. An object in the data service that contains the FieldMappingRow associated with this FieldMappingField. Specifying a non-existent object causes an error.
`priority`	int	Required. Represents the priority that the data service uses when it updates the field, relative to other update rules for the same field. Valid values are 1-100.

Declarative Metadata Sample Definition

The following is an example of a CleanDataService component for the lead standard object.

```
<?xml version="1.0" encoding="UTF-8"?>
<CleanDataService xmlns="http://soap.sforce.com/2006/04/metadata">
    <cleanRules>
        <bulkEnabled>false</bulkEnabled>
        <bypassTriggers>false</bypassTriggers>
        <bypassWorkflow>false</bypassWorkflow>
        <description>Adds data info to leads</description>
        <developerName>DataService_Leads_Enrichment</developerName>
        <fieldMappings>
            <SObjectType>DataServiceCompanyObject</SObjectType>
            <developerName>DataService_Leads_Enrichment_InputMapping</developerName>
            <fieldMappingRows>
                <SObjectType>DataServiceCompanyObject</SObjectType>
                <fieldMappingFields>
```

```
                <dataServiceField>Email</dataServiceField>
                <dataServiceObjectName>Lead</dataServiceObjectName>
                <priority>1</priority>
            </fieldMappingFields>
            <fieldName>Email</fieldName>
            <mappingOperation>Autofill</mappingOperation>
        </fieldMappingRows>
        <fieldMappingRows>
            <SObjectType>DataServiceCompanyObject</SObjectType>
            <fieldMappingFields>
                <dataServiceField>Company</dataServiceField>
                <dataServiceObjectName>Lead</dataServiceObjectName>
                <priority>1</priority>
            </fieldMappingFields>
            <fieldName>Name</fieldName>
            <mappingOperation>Autofill</mappingOperation>
        </fieldMappingRows>
        <masterLabel>DataServiceInputMapping</masterLabel>
    </fieldMappings>
    <fieldMappings>
        <SObjectType>Lead</SObjectType>
        <developerName>DataService_Leads_Enrichment_OutputMapping</developerName>
        <fieldMappingRows>
            <SObjectType>Lead</SObjectType>
            <fieldMappingFields>
                <dataServiceField>EmployeesTotal</dataServiceField>
              <dataServiceObjectName>DataServiceCompanyObject</dataServiceObjectName>

                <priority>1</priority>
            </fieldMappingFields>
            <fieldName>NumberOfEmployees</fieldName>
            <mappingOperation>Autofill</mappingOperation>
        </fieldMappingRows>
        <fieldMappingRows>
            <SObjectType>Lead</SObjectType>
            <fieldMappingFields>
                <dataServiceField>Revenue</dataServiceField>
              <dataServiceObjectName>DataServiceCompanyObject</dataServiceObjectName>

                <priority>1</priority>
            </fieldMappingFields>
            <fieldName>AnnualRevenue</fieldName>
            <mappingOperation>Autofill</mappingOperation>
        </fieldMappingRows>
        <fieldMappingRows>
            <SObjectType>Lead</SObjectType>
            <fieldMappingFields>
                <dataServiceField>Industry</dataServiceField>
              <dataServiceObjectName>DataServiceCompanyObject</dataServiceObjectName>

                <priority>1</priority>
            </fieldMappingFields>
            <fieldName>Industry</fieldName>
            <mappingOperation>Autofill</mappingOperation>
```

```
            </fieldMappingRows>
            <masterLabel>DataServiceOutputMapping</masterLabel>
        </fieldMappings>
        <masterLabel>Data Service Company Info for Leads</masterLabel>
        <matchRule>DataServiceLeadAppendMatchRule</matchRule>
        <sourceSobjectType>DataServiceCompanyObject</sourceSobjectType>
        <status>Active</status>
        <targetSobjectType>Lead</targetSobjectType>
    </cleanRules>
    <description>Data Service Companies for Leads</description>
    <masterLabel>Data Service Companies for Leads</masterLabel>
    <matchEngine>LeadEnrichmentMatchEngine</matchEngine>
</CleanDataService>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>DataService_Leads_Enrichment</members>
        <name>CleanDataService</name>
    </types>
    <version>38.0</version>
</Package>
```

Usage

Use CleanDataService to retrieve all the metadata that describes a data enrichment service. To configure the service in a new org, deploy the metadata you retrieved. Avoid using CRUD-Based Calls with CleanDataService.

To make small modifications to the CleanDataService component, use the Tooling API.

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

CMSConnectSource

Represents the connection information for external content management systems that feed content to Experience Builder sites. This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Note: For use with Change Sets, CMSConnectSource is a dependent of **Network** and **Community**.

File Suffix and Directory Location

CMSConnectSource components have the suffix `.cmsConnectSource` and are stored in the `cmsConnectSource` folder. In that folder, separate files exist for each network (for example, *`networkname`*`.`*`sourcedevelopername`*`.cmsConnectSource`). Each file represents a CMS connection.

Version

CMSConnectSource components are available in API version 43.0 and later.

Special Access Rules

The **CMS Connect org** permission must be enabled.

Fields

Field Name	Field Type	Description
`cmsConnectAsset`	CMSConnectAsset on page 519[]	Represents CSS or JavaScript defined for the connection. • 0–10 for CSS • 0–10 for JavaScript
`cmsConnectLanguage`	CMSConnectLanguage on page 520[]	0 to more. Represents language mappings defined for the connection.
`cmsConnectPersonalization`	CMSConnectPersonalization[] on page 520	0 or 1. Represents personalization defined for the connection. Only for use when `type` is `AEM`.
`cmsConnectResourceType`	CMSConnectResourceType on page 520[]	0–5. Represents JSON definitions defined for the connection.
`connectionType`	CMSSourceConnectionType (enumeration of type string)	Required. Type of authentication being used with outside system. Valid values are: • `Public` • `Authenticated`
`cssScope`	string	The class name used to prefix and scope the CSS rules.
`developerName`	string	Required. API name of the CMSConnectSource entity.
`languageEnabled`	string	Required. Valid values are: • `Y` to enable language mapping for connection. • `N` if no language mapping is needed.
`masterLabel`	string	Required. Connection name
`namedCredential`	string	Required when the `connectionType` is `Authenticated`. API name of `namedCredential`. Before deploying `namedCredential`, it must exist on the destination org.

Field Name	Field Type	Description
`personalizationEnabled`	string	Required. Valid values are: • `Y` to enable personalization mapping for connection. • Otherwise `N`.
`rootPath`	string	Required. Root path.
`sortOrder`	int	Required. Defines the load order of the connection when multiple connections defined on page. The load order begins with 1.
`status`	CMSConnectionStatus (enumeration of type string)	Required. Status of connection. Valid values are: • `ACTIVE` • `INACTIVE`
`type`	CMSConnectionSourceType (enumeration of type string)	Required. The identification of the source connection system. Valid values are: • `AEM` • `Drupal` • `WordPress` • `SDL` • `Sitecore` • `Other`
`websiteUrl`	string	Required if `connectionType` is `Public`

Note: Because there can be existing connections when a package comes in, there's some INSERT or UPDATE logic to consider:

- If you find `developerName` in the destination, then update the existing collection with all details form source.
- `namedCredential` is handled through `developerName`. If you don't find `namedCredential` with `developerName`, then an error is generated.
- If the destination isn't `sortOrder` from the source, then insert or update with the source `sortOrder`.
- If `sortOrder` from the source is already in the destination, then increase the source `sortOrder` by 1 for connections such that the destination `sortOrder` > `sortOrder` from the source.

CMSConnectAsset

CMSConnectAsset defines the location, types, and order of assets necessary to support the incoming content, such as JavaScript and CSS files.

Note: Because there can be existing connections when a package comes in, there's some INSERT or UPDATE logic to consider:

- If `assetPath` exists in the destination, then update the existing record, else the new `assetPath` is inserted.
- Always keep the `sortOrder` from the source and adjust the destination accordingly.

Field Name	Field Type	Description
`assetPath`	string	Relative path of the asset.
`assetType`	string	When used in Apex, this value can be sent as an enum, otherwise, this field has a type of string. Allowed values as string • `CSS` • `Javascript` Allowed values as enum • `CSS` • `Javascript`
`sortOrder`	int	Loading sequence on the page.

CMSConnectLanguage

CMSConnectLanguage components determine the presented language of the content.

Field Name	Field Type	Description
`cmsLanguage`	string	When a language placeholder is in the URL path, this value is used to replace it.
`language`	string	Salesforce supported language. For information see https://developer.salesforce.com/docs/atlas.en-us.api_meta.meta/api_meta/meta_translations.htm

CMSConnectPersonalization

CMSConnectPersonalization is used only with Adobe Experience Manager (AEM).

Note: Because there can be existing connections when a package comes in, there's some INSERT or UPDATE logic to consider. If personalization isn't enabled in the source system, but is enabled in the destination, the destination is disabled. The record for the connection is deleted from the table.

Field Name	Field Type	Description
`connectorPage`	string	The path to the JSP file that you created and installed in AEM.
`connectorPageAsset`	string	The path to your Javascript file. Providing this path allows you to run scripts dynamically.

CMSConnectResourceType

CMSConnectResourceType is for use only to define JSON connections.

Note: Because there can be existing connections when a package comes in, there's some INSERT or UPDATE logic to consider. If you find the developer name in the destination, then update the existing record with all details from the source.

Field Name	Field Type	Description
cmsConnectResourceDefinition	cmsConnectResourceDefinition on page 521[]	0–10 allowed per CMSConnectResourceType.
developerName	string	API name of CMSConnectResourceType.
masterLabel	string	Content type name.
resourceType	string	The only allowed value is JSON.

CMSConnectResourceDefinition

cmsConnectResourceDefinition is used to define JSON connections.

Note: Because there can be existing connections when a package comes in, there's some INSERT or UPDATE logic to consider:

- If you find developerName in the destination, then the existing record is updated with all details from the new source, else the new value is inserted.
- If the current source is DETAIL and the destination has DETAIL with a different name, then the destination is updated to LIST and the source is inserted as DETAIL.

Field Name	Field Type	Description
developerName	string	Required. API name of CMSConnectResourceDefinition.
masterLabel	string	Required. developerName of Content Item or Content List.
options	int	Required. Identifies whether the content from the external source is a single item or a list. 0 for Content List 1 for Content Item
payloadType	string	Required. The only valid value is JSON.
resourceIdPath	string	Relative path to ID. Required for Content Item.
resourceNamePath	string	Relative path to resource name. Required for Content Item.
resourcePath	string	Required. JSON resource path.
rootNodePath	string	Only for Content List and collection. Defines the initial starting path for a collection or list.

Declarative Metadata Sample Definition

The following is an example of a CMSConnectSource definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<CMSConnectSource xmlns="http://soap.sforce.com/2006/04/metadata">
    <cmsConnectAsset>
        <assetPath>etc/designs/capricorn/app-prefixed.min.css</assetPath>
        <assetType>CSS</assetType>
        <sortOrder>1</sortOrder>
    </cmsConnectAsset>
    <cmsConnectAsset>
        <assetPath>etc/designs/capricorn/w3data.js</assetPath>
        <assetType>Javascript</assetType>
        <sortOrder>1</sortOrder>
    </cmsConnectAsset>
    <cmsConnectLanguage>
        <cmsLanguage>en</cmsLanguage>
        <language>en_US</language>
    </cmsConnectLanguage>
    <cmsConnectLanguage>
        <cmsLanguage>fr</cmsLanguage>
        <language>fr</language>
    </cmsConnectLanguage>
    <cmsConnectPersonalization>
        <connectorPage>content/salesforceConnector.js</connectorPage>
        <connectorPageAsset>content/js/capricorn/assets.js</connectorPageAsset>
    </cmsConnectPersonalization>
    <cmsConnectResourceType>
        <cmsConnectResourceDefinition>
            <developerName>Details</developerName>
            <masterLabel>Details</masterLabel>
            <options>0</options>
            <payloadType>JSON</payloadType>
            <resourceIdPath>ID</resourceIdPath>
            <resourceNamePath>title</resourceNamePath>

<resourcePath>rest/v1.1/sites/cmstry.wordpress.com/posts/{component}</resourcePath>
        </cmsConnectResourceDefinition>
        <cmsConnectResourceDefinition>
            <developerName>List</developerName>
            <masterLabel>List</masterLabel>
            <options>1</options>
            <payloadType>JSON</payloadType>

<resourcePath>rest/v1.1/sites/cmstry.blog.wordpress.com/posts?number={itemsPerPage}&amp;amp;page={pageNumber}</resourcePath>

        </cmsConnectResourceDefinition>
        <developerName>Posts</developerName>
        <masterLabel>Posts</masterLabel>
        <resourceType>JSON</resourceType>
    </cmsConnectResourceType>
    <connectionType>Public</connectionType>
    <cssScope>capricorn</cssScope>
    <developerName>Capricorn</developerName>
```

```
    <languageEnabled>Y</languageEnabled>
    <masterLabel>Capricorn</masterLabel>
    <personalizationEnabled>Y</personalizationEnabled>
    <rootPath>content/capricorn/{language}</rootPath>
    <sortOrder>11</sortOrder>
    <status>ACTIVE</status>
    <type>AEM</type>
    <websiteUrl>https://public-api.wordpress.com</websiteUrl>
</CMSConnectSource>
```

The following is an example `package.xml`.

```
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>NetworkA.*</members>
        <name>CMSConnectSource</name>
    </types>
    <version>43.0</version>
</Package>
```

To retrieve a specific connection:

```
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>NetworkA.DeveloperName</members>
        <name>CMSConnectSource</name>
    </types>
    <version>43.0</version>
</Package>
```

Usage

The INSERT or UPDATE logic for the incoming information is always UPSERT. If data isn't in the entity, then it's inserted, otherwise the current data is updated.

Before doing upsert, the content from the package is validated against the maximum limits for the following:

- CSS assets <= 10
- JavaScript assets <= 10
- Resource types < =5
- Resource definitions for each type <=10

For example

1. The validation on a new connection totals only the elements in the incoming package.
2. Validation of existing connections totals the existing assets and new elements to assess validity. For example, if a connection on the destination org already has six CSS definitions, and the incoming package has defined seven CSS definitions (four new + three existing), the new total is the six current from the database. The total ignores the three repeated in the package and adds four new definitions from the incoming package. This totals 10 definitions, which number is at or below the 10 asset threshold, and it passes validation.

Refer to the following content for more details for how each entity how is handled while saving the details from package to destination org:

Type	Description
`CMSConnectSource`	• If you find `developerName` in the destination, then update the existing collection with all details form source. • `namedCredential` is handled through `developerName`. If you don't find `namedCredential` with `developerName`, then an error is generated. • If the destination isn't `sortOrder` from the source, then insert or update with the source `sortOrder`. • If `sortOrder` from the source is already in the destination, then increase the source `sortOrder` by 1 for connections such that the destination `sortOrder` > `sortOrder` from the source.
`CMSConnectAsset`	• If `assetPath` exists in the destination, then update the existing record, else the new `assetPath` is inserted. • Always keep the `sortOrder` from the source and adjust the destination accordingly.
`CMSConnectPersonalization`	If personalization isn't enabled in the source system, but is enabled in the destination, the destination is disabled. The record for the connection is deleted from the table.
`CMSConnectResourceType`	If you find the developer name in the destination, then update the existing record with all details from the source.
`CMSConnectResourceDefinition`	• If you find developerName in the destination, then the existing record is updated with all details from the new source, else the new value is inserted. • If the current source is DETAIL and the destination has DETAIL with a different name, then the destination is updated to LIST and the source is inserted as DETAIL.

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SEE ALSO:

- Select Components for an Outbound Change Set
- View and Add Dependent Components to a Change Set
- Developer Guide: Deploying and Retrieving Metadata
- Salesforce Help: Use Personalized Content in CMS Connect
- Developer Guide: Translations

Community (Zone)

Represents a zone that contains Ideas or Chatter Answers objects. Zones are shared by the Ideas, Answers, and Chatter Answers features, allowing you to view and create zones from those locations.This type extends the Metadata metadata type and inherits its `fullName` field.

Note: Starting with the Summer '13 release, Chatter Answers and Ideas "communities" have been renamed to "zones." In API version 28, the API object label has changed to `Zone`, but the API type is still `Community`.

File Suffix and Directory Location

Zones have the suffix `community` and are stored in the `communities` folder.

Version

Community (Zone) components are available in API version 27.0 and later.

Fields

Note: When `enableChatterAnswers` is set to `false`, values specified for the following fields are ignored and not saved: `communityFeedPage`, `emailFooterDocument`, `emailHeaderDocument`, `enablePrivateQuestions`, `emailNotificationUrl`, and `site`.

Field Name	Field Type	Description
`active`	boolean	Indicates whether the zone is active (`true`) or not (`false`).
`chatterAnswersFacebookSsoUrl`	string	(Read only) The Facebook sign-on URL, which is based on the Facebook authentication provider selected in your Chatter Answers settings. This field is available only if Chatter Answers and Facebook Single Sign-On for Chatter Answers are enabled.
`communityFeedPage`	string	The Visualforce page that hosts the zone's feeds. This field is available when Chatter Answers is enabled in the organization.
`description`	string	The description of the zone.
`emailFooterDocument`	string	The text or HTML file that incorporates your organization's branding into the footer of email notifications. This field is available when Chatter Answers is enabled in the organization.
`emailHeaderDocument`	string	The text or HTML file that incorporates your organization's branding into the header of email notifications. This field is available when Chatter Answers is enabled in the organization.
`emailNotificationUrl`	string	The URL that's included in email notifications. This field is available when Chatter Answers is enabled in the organization. This field replaces `portalEmailNotificationUrl` in API version 28.0 and later.

Field Name	Field Type	Description
enableChatterAnswers	boolean	Indicates whether the zone has Chatter Answers enabled (true) or not (false). This field is available when Chatter Answers is enabled in the organization.
enablePrivateQuestions	boolean	Indicates whether Chatter Answers questions can be escalated to cases (true) or not (false). This field is available when Chatter Answers is enabled in the organization.
expertsGroup	string	The name of the public group that act as experts in the zone. This field is available when either Ideas or Answers are enabled in the organization.
portal	string	The name of the portal in which to display the zone.
portalEmailNotificationUrl	string	The portal URL that's included in email notifications. This field is available when Chatter Answers is enabled in the organization. This field has been replaced by emailNotificationUrl in API version 28.0 and later.
reputationLevels	ReputationLevels	The fields that define the points and name of each reputation level you define. You can create up to 25 reputation levels per zone.
showInPortal	boolean	Indicates whether the zone is available to all portals (true) or not available to any portals (false).
site	string	The name of the site for the zone. This field is available when Chatter Answers is enabled in the organization.

ReputationLevels

Represents the points and reputation label that displays on hover over a user's photo in the feed.

Field Name	Field Type	Description
chatterAnswersReputationLevels	ChatterAnswersReputationLevel []	Contains the name and value pair that describes the reputation level for Chatter Answers. Available in API version 28.0 and later.
ideaReputationLevels	IdeaReputationLevel	Contains the name and value pair that describes the reputation for Ideas. Available in API version 28.0 and later.

ChatterAnswersReputationLevel

Represents the reputation name and the number of points for that level for Chatter Answers.

Field Name	Field Type	Description
name	string	The name of the reputation level, for example, "Expert."
value	int	The minimum number of points for the reputation level.

IdeaReputationLevel

Represents the reputation name and the number of points for that level for Ideas. Available in API version 28.0 and later.

Field Name	Field Type	Description
name	string	The name of the reputation level, for example, “Expert.”
value	int	The minimum number of points for the reputation level.

Declarative Metadata Sample Definition

The following is the definition of a community (zone) component:

```
<?xml version="1.0" encoding="UTF-8"?>
<Community xmlns="http://soap.sforce.com/2006/04/metadata">
    <active>true</active>
    <communityFeedPage>communityWithHeaderAndFooter_main</communityFeedPage>
    <description>Metadata Test</description>
    <emailFooterDocument>sampleFolder/emailFooter.html</emailFooterDocument>
    <emailHeaderDocument>sampleFolder/emailHeader.html</emailHeaderDocument>
    <enableChatterAnswers>true</enableChatterAnswers>
    <enablePrivateQuestions>true</enablePrivateQuestions>
    <expertsGroup>CommunityExperts</expertsGroup>
    <portal>Customer Portal</portal>
    <emailNotificationUrl>http://yourURL</emailNotificationUrl>
<reputationLevels>
        <chatterAnswersReputationLevels>
            <name>Newbie</name>
            <value>0</value>
        </chatterAnswersReputationLevels>
        <chatterAnswersReputationLevels>
            <name>Smartie</name>
            <value>500</value>
        </chatterAnswersReputationLevels>
        <chatterAnswersReputationLevels>
            <name>Pro</name>
            <value>2000</value>
        </chatterAnswersReputationLevels>
        <chatterAnswersReputationLevels>
            <name>All Star</name>
            <value>5000</value>
        </chatterAnswersReputationLevels>
        <ideaReputationLevels>
            <name>Observer</name>
            <value>0</value>
        </ideaReputationLevels>
        <ideaReputationLevels>
            <name>Contributor</name>
            <value>100</value>
        </ideaReputationLevels>
        <ideaReputationLevels>
            <name>Influencer</name>
```

```
            <value>400</value>
        </ideaReputationLevels>
        <ideaReputationLevels>
            <name>Thought Leader</name>
            <value>1500</value>
        </ideaReputationLevels>
    </reputationLevels>
    <showInPortal>true</showInPortal>
    <site>ChatterAnswersSite</site>
</Community>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

CommerceSettings

Represents settings for various Commerce features.

Parent Type and Manifest Access

This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all the settings metadata types for the org are accessed using the "Settings" name. See Settings for more details.

File Suffix and Directory Location

CommerceSettings values are stored in the `Commerce.settings` file in the `settings` folder. The `.settings` files are different from other named components, because there's only one settings file for each settings component.

Version

Commerce Settings are available in API version 50.0 and later.

Special Access Rules

A B2B Commerce or D2C Commerce license and access to Commerce objects is required.

Fields

Field Name	Description
`commerceAppEnabled`	**Field Type** boolean

Field Name	Description
	Description Indicates whether Commerce App is enabled (`true`) or not (`false`).
`commerceAnalyticsEnabled`	**Field Type** boolean **Description** Indicates whether Commerce Analytics is enabled (`true`) or not (`false`).
`commerceConciergeEnabled`	**Field Type** boolean **Description** Indicates whether Commerce Concierge bots are enabled (`true`) or not (`false`).
`commerceCopilotEcomEnabled`	**Field Type** boolean **Description** Indicates whether Commerce Copilot is enabled (`true`) or not (`false`).
`commerceDiscoveryExpansion`	**Field Type** boolean **Description** Indicates whether the Commerce Discovery Expansion service is enabled (`true`) or not (`false`).
`commerceDCSegmentEnabled`	**Field Type** boolean **Description** Indicates whether the Data Cloud segment integration is enabled (`true`) or not (`false`).
`commerceEnabled`	**Field Type** boolean **Description** Indicates whether Commerce is enabled (`true`) or not (`false`).
`commerceNGPEnabled`	**Field Type** boolean **Description** Indicates whether NGP (“Salesforce”) Pricing is enabled (`true`) or not (`false`).
`commerceRLMSubs`	**Field Type** boolean

Field Name	Description
	Description Indicates whether Commerce Revenue Lifecycle Management Subscriptions is enabled (`true`) or not (`false`).
`lowestUnitPriceTracking`	**Field Type** boolean **Description** Indicates whether lowest unit price tracking (for EU customers) is enabled (`true`) or not (`false`).
`messagingEngagementDataKit`	**Field Type** boolean **Description** Indicates whether Message Engagement data kit is enabled (`true`) or not (`false`). Message Engagement data kit is a Data Cloud data model object (DMO) for a user's engagement with a marketing message.

Declarative Metadata Sample Definition

The following is an example of a CommerceSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<CommerceSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <commerceEnabled>false</commerceEnabled>
</CommerceSettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>Commerce</members>
        <name>Settings</name>
    </types>
    <version>60.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

CommunityTemplateDefinition

Represents the definition of an Experience Builder site template. This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

CommunityTemplateDefinition components have the suffix `.communityTemplateDefinition` and are stored in the `communityTemplateDefinitions` folder.

Version

CommunityTemplateDefinition components are available in API version 38.0 and later.

Special Access Rules

This type is available only if Salesforce Digital Experiences is enabled in your org.

Fields

Field Name	Field Type	Description
`baseTemplate`	CommunityBase Template (enumeration of type string)	Denotes that this CommunityTemplateDefinition was created in API version 41.0 or later. The only valid value is `c`. This field is available in API 41.0 and later.
`bundlesInfo`	CommunityTemplateBundleInfo[]	The list of preview images and feature highlights of this CommunityTemplateDefinition.
`category`	CommunityTemplate Category (enumeration of type string)	Required. The optimized use case of this CommunityTemplateDefinition. Valid values are: • `Commerce` • `IT` • `Marketing` • `Sales` • `Service`
`defaultBrandingSet`	string	The set of branding properties associated with this CommunityTemplateDefinition, as defined in the Theme panel in Experience Builder. Available in API version 40.0 and later. In API version 44.0 and later, this field is read-only and can be implemented in CommunityThemeDefinition on page 537.

Field Name	Field Type	Description
`defaultThemeDefinition`	string	Required. The assigned theme definition for this CommunityTemplateDefinition.
`description`	string	The optional description text of this CommunityTemplateDefinition.
`enableExtendedCleanUpOnDelete`	boolean	False by default. Determines if deleting this CommunityTemplateDefinition attempts to delete other directly or indirectly referenced objects automatically, for example, CommunityThemeDefinition on page 537, Flexipage on page 1090, or StaticResource on page 2185. Values are true or false.
`masterLabel`	string	Required. The label for this CommunityTemplateDefinition, which displays in Setup.
`navigationLinkSet`	NavigationLinkSet	The navigation menu associated with this CommunityTemplateDefinition. A navigation menu consists of items that users can click to go to other parts of the site. Available in API versions 37.0 to 46.0. In API versions 47.0 and later, use NavigationMenu.
`pageSetting`	CommunityTemplatePageSetting[]	The list of FlexiPage of this CommunityTemplateDefinition.
`publisher`	string	Defines the name of the publisher as seen in the Change Theme wizard. If no name is provided, the name of the org from which the package was originally exported is used. This field is available in API version 45.0 and later.

CommunityTemplateBundleInfo

Field Name	Field Type	Description
`description`	string	The optional description text of its CommunityTemplateBundleInfo.
`image`	string	Required only when the `type` is `PreviewImage`, otherwise this field is optional. A preview image for this CommunityTemplateDefinition.
`order`	int	Required. An integer specifying the position of this CommunityTemplateBundleInfo relative to others of the same `type` within its CommunityTemplateDefinition. `1` is the first position, `3` is the maximum position for `PreviewImage` type, and `4` is the maximum position for the `Highlight` type.
`title`	string	Required. The title of this CommunityTemplateBundleInfo to use in code.
`type`	CommunityTemplateBundleInfoType (enumeration of type string)	Required. Stores descriptive information about the template that's included in the export. The template powers the interface of the Experience Creation Wizard. Valid values are: • `Highlight`—This CommunityTemplateBundleInfo is used as a highlighted feature. Up to 4 are supported.

Field Name	Field Type	Description
		• `PreviewImage`—This CommunityTemplateBundleInfo is used as a preview image. Up to 3 are supported.

CommunityTemplatePageSetting

Field Name	Field Type	Description
`page`	string	Required. The list of FlexiPage of this CommunityTemplateDefinition.
`themeLayout`	string	Required. The name of the FlexiPage for the theme layout. This field is available in API version 39.0 and later.

Declarative Metadata Sample Definition

The following is an example of a CommunityTemplateDefinition component.

```
<?xml version="1.0" encoding="UTF-8"?>
<CommunityTemplateDefinition xmlns="http://soap.sforce.com/2006/04/metadata">
    <baseTemplate>c</baseTemplate>
    <bundlesInfo>
        <description>Feature Description</description>
        <order>1</order>
        <title>Feature Heading</title>
        <type>Highlight</type>
    </bundlesInfo>
    <bundlesInfo>
        <image>siteAsset_2dbe594eb6794173af78da264cd6a4a7</image>
        <order>1</order>
        <title>Preview Image</title>
        <type>PreviewImage</type>
    </bundlesInfo>
    <category>Sales</category>
    <defaultThemeDefinition>communityTemplate</defaultThemeDefinition>
    <description>This is an Experience Builder template</description>
    <enableExtendedCleanUpOnDelete>true</enableExtendedCleanUpOnDelete>
    <masterLabel>communityTemplate</masterLabel>
    <navigationLinkSet>
        <navigationMenuItem>
            <label>Topics</label>
            <position>0</position>
            <publiclyAvailable>true</publiclyAvailable>
            <target>ShowMoreTopics</target>
            <type>NavigationalTopic</type>
        </navigationMenuItem>
    </navigationLinkSet>
    <pageSetting>
        <page>communityTemplate_Report_List</page>
        <themeLayout>communityTemplate_themeLayout_Default</themeLayout>
```

```
    </pageSetting>
    <pageSetting>
        <page>communityTemplate_Topic_Catalog</page>
        <themeLayout>communityTemplate_themeLayout_Default</themeLayout>
    </pageSetting>
    <pageSetting>
        <page>communityTemplate_Check_Password</page>
        <themeLayout>communityTemplate_themeLayout_Login</themeLayout>
    </pageSetting>
    <pageSetting>
        <page>communityTemplate_Error</page>
        <themeLayout>communityTemplate_themeLayout_Default</themeLayout>
    </pageSetting>
    <pageSetting>
        <page>communityTemplate_User_Settings</page>
        <themeLayout>communityTemplate_themeLayout_Default</themeLayout>
    </pageSetting>
    <pageSetting>
        <page>communityTemplate_Login</page>
        <themeLayout>communityTemplate_themeLayout_Login</themeLayout>
    </pageSetting>
    <pageSetting>
        <page>communityTemplate_Stream_List</page>
        <themeLayout>communityTemplate_themeLayout_Default</themeLayout>
    </pageSetting>
    <pageSetting>
        <page>communityTemplate_Sfdc_Page</page>
        <themeLayout>communityTemplate_themeLayout_Default</themeLayout>
    </pageSetting>
    <pageSetting>
        <page>communityTemplate_Group_Detail</page>
        <themeLayout>communityTemplate_themeLayout_Default</themeLayout>
    </pageSetting>
    <pageSetting>
        <page>communityTemplate_Report_Related_List</page>
        <themeLayout>communityTemplate_themeLayout_Default</themeLayout>
    </pageSetting>
    <pageSetting>
        <page>communityTemplate_Register</page>
        <themeLayout>communityTemplate_themeLayout_Login</themeLayout>
    </pageSetting>
    <pageSetting>
        <page>communityTemplate_User_Profile</page>
        <themeLayout>communityTemplate_themeLayout_Default</themeLayout>
    </pageSetting>
    <pageSetting>
        <page>communityTemplate_Case_Detail</page>
        <themeLayout>communityTemplate_themeLayout_Default</themeLayout>
    </pageSetting>
    <pageSetting>
        <page>communityTemplate_Stream_Related_List</page>
        <themeLayout>communityTemplate_themeLayout_Default</themeLayout>
    </pageSetting>
    <pageSetting>
```

```
        <page>communityTemplate_Dashboard_Detail</page>
        <themeLayout>communityTemplate_themeLayout_Default</themeLayout>
    </pageSetting>
    <pageSetting>
        <page>communityTemplate_Group_List</page>
        <themeLayout>communityTemplate_themeLayout_Default</themeLayout>
    </pageSetting>
    <pageSetting>
        <page>communityTemplate_Canvasapp_Page</page>
        <themeLayout>communityTemplate_themeLayout_Default</themeLayout>
    </pageSetting>
    <pageSetting>
        <page>communityTemplate_Login_Error</page>
        <themeLayout>communityTemplate_themeLayout_Login</themeLayout>
    </pageSetting>
    <pageSetting>
        <page>communityTemplate_Create_Record</page>
        <themeLayout>communityTemplate_themeLayout_Default</themeLayout>
    </pageSetting>
    <pageSetting>
        <page>communityTemplate_Group_Related_List</page>
        <themeLayout>communityTemplate_themeLayout_Default</themeLayout>
    </pageSetting>
    <pageSetting>
        <page>communityTemplate_Search</page>
        <themeLayout>communityTemplate_themeLayout_Default</themeLayout>
    </pageSetting>
    <pageSetting>
        <page>communityTemplate_File_Detail</page>
        <themeLayout>communityTemplate_themeLayout_Default</themeLayout>
    </pageSetting>
    <pageSetting>
        <page>communityTemplate_Case_List</page>
        <themeLayout>communityTemplate_themeLayout_Default</themeLayout>
    </pageSetting>
    <pageSetting>
        <page>communityTemplate_User_List</page>
        <themeLayout>communityTemplate_themeLayout_Default</themeLayout>
    </pageSetting>
    <pageSetting>
        <page>communityTemplate_File_List</page>
        <themeLayout>communityTemplate_themeLayout_Default</themeLayout>
    </pageSetting>
    <pageSetting>
        <page>communityTemplate_Question_Detail</page>
        <themeLayout>communityTemplate_themeLayout_Default</themeLayout>
    </pageSetting>
    <pageSetting>
        <page>communityTemplate_Dashboard_List</page>
        <themeLayout>communityTemplate_themeLayout_Default</themeLayout>
    </pageSetting>
    <pageSetting>
        <page>communityTemplate_Related_Record_List</page>
        <themeLayout>communityTemplate_themeLayout_Default</themeLayout>
```

```
    </pageSetting>
    <pageSetting>
        <page>communityTemplate_File_Related_List</page>
        <themeLayout>communityTemplate_themeLayout_Default</themeLayout>
    </pageSetting>
    <pageSetting>
        <page>communityTemplate_Record_List</page>
        <themeLayout>communityTemplate_themeLayout_Default</themeLayout>
    </pageSetting>
    <pageSetting>
        <page>communityTemplate_Forgot_Password</page>
        <themeLayout>communityTemplate_themeLayout_Login</themeLayout>
    </pageSetting>
    <pageSetting>
        <page>communityTemplate_Home</page>
        <themeLayout>communityTemplate_themeLayout_Default</themeLayout>
    </pageSetting>
    <pageSetting>
        <page>communityTemplate_Dashboard_Related_List</page>
        <themeLayout>communityTemplate_themeLayout_Default</themeLayout>
    </pageSetting>
    <pageSetting>
        <page>communityTemplate_Account_Management</page>
        <themeLayout>communityTemplate_themeLayout_Default</themeLayout>
    </pageSetting>
    <pageSetting>
        <page>communityTemplate_Case_Related_List</page>
        <themeLayout>communityTemplate_themeLayout_Default</themeLayout>
    </pageSetting>
    <pageSetting>
        <page>communityTemplate_User_Related_List</page>
        <themeLayout>communityTemplate_themeLayout_Default</themeLayout>
    </pageSetting>
    <pageSetting>
        <page>communityTemplate_Stream_Detail</page>
        <themeLayout>communityTemplate_themeLayout_Default</themeLayout>
    </pageSetting>
    <pageSetting>
        <page>communityTemplate_Topic_Detail</page>
        <themeLayout>communityTemplate_themeLayout_Default</themeLayout>
    </pageSetting>
    <pageSetting>
        <page>communityTemplate_Messages</page>
        <themeLayout>communityTemplate_themeLayout_Default</themeLayout>
    </pageSetting>
    <pageSetting>
        <page>communityTemplate_Report_Detail</page>
        <themeLayout>communityTemplate_themeLayout_Default</themeLayout>
    </pageSetting>
    <pageSetting>
        <page>communityTemplate_Record_Detail</page>
        <themeLayout>communityTemplate_themeLayout_Default</themeLayout>
    </pageSetting>
    <pageSetting>
```

```
        <page>communityTemplate_Feed_Detail</page>
        <themeLayout>communityTemplate_themeLayout_Default</themeLayout>
    </pageSetting>
    <pageSetting>
        <page>communityTemplate_Contact_Support</page>
        <themeLayout>communityTemplate_themeLayout_Default</themeLayout>
    </pageSetting>
</CommunityTemplateDefinition>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>MyTemplate</members>
        <name>CommunityTemplateDefinition</name>
    </types>
    <version>64.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

CommunityThemeDefinition

Represents the definition of a theme for an Experience Builder site. This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

CommunityThemeDefinition components have the suffix `.communityThemeDefinition` and are stored in the `communityThemeDefinitions` folder.

Version

CommunityThemeDefinition components are available in API version 38.0 and later.

Special Access Rules

This type is available only if Salesforce Digital Experiences is enabled in your org.

Fields

Field Name	Field Type	Description
bundlesInfo	CommunityThemeBundleInfo[]	If specified, at least one preview image and one highlight are required. Up to 3 preview images and 4 highlights are supported. Available in API version 44.0 and later
customThemeLayoutType	CommunityCustomThemeLayoutType[]	The list of custom theme layout types available to the theme layout.
defaultBrandingSet	string	The set of branding properties associated with this CommunityThemeDefinition, as defined in the Theme panel in Experience Builder. Available in API version 44.0 and later.
description	string	The optional description text of this CommunityThemeDefinition.
enableExtendedCleanUpOnDelete	boolean	False by default. Determines if deleting this CommunityThemeDefinition attempts to delete other directly or indirectly referenced objects automatically, for example, FlexiPage. Values are true or false.
masterLabel	string	Required. The label for this CommunityThemeDefinition, which displays in Setup.
publisher	string	Defines the name of the publisher as seen in the wizard for creating Experience Builder sites. If no name is provided, the name of the org from which the package was originally exported is used. This field is available in API version 45.0 and later.
themeRouteOverride	CommunityThemeRouteOverride[]	List of theme layout type overrides for flexipages (currently only for home). Available in API version 44.0 and later.
themeSetting	CommunityThemeSetting []	Required. The list of settings for this CommunityThemeDefinition.

CommunityThemeBundleInfo

Field Name	Field Type	Description
description	string	The optional description text of its CommunityThemeBundleInfo.
image	string	Required only when the `type` is `PreviewImage`, otherwise this field is optional. A preview image for this CommunityThemeDefinition.
order	int	Required. An integer specifying the position of this CommunityThemeBundleInfo relative to others of the same `type` within its CommunityThemeDefinition. `1` is the first position, `3` is the maximum position for `PreviewImage` type, and `4` is the maximum position for the `Highlight` type.
title	string	Required. The title of this CommunityThemeBundleInfo to use in code.

Field Name	Field Type	Description
`type`	CommunityTemplate BundleInfoType (enumeration of type string)	Required. Stores descriptive information about the theme that is included in the export. Valid values are: • `Highlight`—This CommunityThemeBundleInfo is used as a highlighted feature. Up to 4 are supported. • `PreviewImage`—This CommunityThemeBundleInfo is used as a preview image. Up to 3 are supported.

CommunityCustomThemeLayoutType

Field Name	Field Type	Description
`description`	string	The description of the custom theme layout type.
`label`	string	Required. The name of the custom theme layout type. The values `Inner`, `Home`, and `Login` are reserved.

CommunityThemeRouteOverride

Field Name	Field Type	Description
`customThemeLayoutType`	string	Required when `themeLayoutType` isn't specified. Provides the custom theme layout type associated with the theme layout. This field and `themeLayoutType` are mutually exclusive; you can't specify both.
`pageAttributes`	string	Required. Specifies the attributes of the site page for which the default theme layout type is overridden. The only valid value is `{"PageName":"Home"}`.
`pageType`	string	Required. Specifies the type of the site page for which the default theme layout type is overridden. The only valid value is `comm__standardPage`.
`themeLayoutType`	CommunityTheme LayoutType (enumeration of type string)	Required if `customThemeLayoutType` isn't specified. Provides the default theme layout type associated with the theme layout. Valid values are `Inner`, `Home`, or `Login`. This field and `customThemeLayoutType` are mutually exclusive; you can't specify both.

CommunityTheme Setting

Field Name	Field Type	Description
`customThemeLayoutType`	string	Required when `themeLayoutType` isn't specified. The custom theme layout type associated with the theme layout. This field and `themeLayoutType` are mutually exclusive; you can't specify both.
`themeLayout`	string	Required. The configuration and layout for this theme.

Field Name	Field Type	Description
`themeLayoutType`	CommunityTheme LayoutType (enumeration of type string)	Required when `customThemeLayoutType` isn't specified. The default theme layout type associated with the theme layout. Valid values are `Inner`, `Home`, or `Login`. This field and `customThemeLayoutType` are mutually exclusive; you can't specify both.

Declarative Metadata Sample Definition

The following is an example of a CommunityThemeDefinition component.

```
<?xml version="1.0" encoding="UTF-8"?>
<CommunityThemeDefinition xmlns="http://soap.sforce.com/2006/04/metadata">
    <bundlesInfo>
        <description>Batman Feature1 description</description>
        <order>1</order>
        <title>Batman Feature1</title>
        <type>Highlight</type>
    </bundlesInfo>
    <bundlesInfo>
        <image>siteAsset_d90e2d5ce4cf4d8899e233c051091246</image>
        <order>1</order>
        <title>siteAsset_d90e2d5ce4cf4d8899e233c051091246</title>
        <type>PreviewImage</type>
    </bundlesInfo>
    <defaultBrandingSet>Batman</defaultBrandingSet>
    <description>Batman theme</description>
    <enableExtendedCleanUpOnDelete>true</enableExtendedCleanUpOnDelete>
    <masterLabel>Batman</masterLabel>
    <themeRouteOverride>
        <pageAttributes>{&quot;PageName&quot;:&quot;Home&quot;}</pageAttributes>
        <pageType>comm__standardPage</pageType>
        <themeLayoutType>Home</themeLayoutType>
    </themeRouteOverride>
    <themeSetting>
        <themeLayout>Batman_themeLayout_Login</themeLayout>
        <themeLayoutType>Login</themeLayoutType>
    </themeSetting>
    <themeSetting>
        <themeLayout>Batman_themeLayout_Home</themeLayout>
        <themeLayoutType>Home</themeLayoutType>
    </themeSetting>
    <themeSetting>
        <themeLayout>Batman_themeLayout_Default</themeLayout>
        <themeLayoutType>Inner</themeLayoutType>
    </themeSetting>
</CommunityThemeDefinition>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
```

```
        <members>Batman</members>
        <name>BrandingSet</name>
    </types>
    <types>
        <members>Batman</members>
        <name>CommunityThemeDefinition</name>
    </types>
    <types>
        <members>Batman_themeLayout_Default</members>
        <members>Batman_themeLayout_Home</members>
        <members>Batman_themeLayout_Login</members>
        <name>FlexiPage</name>
    </types>
    <types>
        <members>siteAsset_d90e2d5ce4cf4d8899e233c051091246</members>
        <name>StaticResource</name>
    </types>
    <version>64.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ConnectedApp

Represents a connected app configuration. A connected app enables an external application to integrate with Salesforce using APIs and standard protocols, such as SAML, OAuth, and OpenID Connect. Connected apps use these protocols to authenticate, authorize, and provide single sign-on (SSO) for external apps. The external apps that are integrated with Salesforce can run on the customer success platform, other platforms, devices, or SaaS subscriptions.

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

ConnectedApp components have the suffix `.connectedApp` and are stored in the `connectedApps` folder.

Version

ConnectedApp components are available in API version 29.0 and later.

Fields

Field Name	Field Type	Description
`attributes`	`canvasConfig`	A custom attribute of the connected app.

Field Name	Field Type	Description
	AppCanvasConfig	The configuration options of the connected app if it's exposed as a canvas app.
`contactEmail`	string	Required. The email address that Salesforce uses to contact you or your support team.
`contactPhone`	string	The phone number for Salesforce to use to contact you.
`description`	string	An optional description for your app.
`iconUrl`	string	Reserved for future use.
`infoUrl`	string	An optional URL for a web page with more information about your app.
`ipRanges`	ConnectedAppIpRange[]	Specifies the ranges of IP addresses that can access the app without requiring the user to authenticate with the connected app.
`label`	string	Required. The name of the app.
`logoUrl`	string	An optional logo for the app. The logo appears with the app's entry in the list of apps and on the consent page the user sees when authenticating. The URL must use HTTPS, and the logo can't be larger than 125 pixels high or 200 pixels wide. The default logo is a cloud.
`mobileStartUrl`	string	Users are directed to this URL after they've authenticated when the app is accessed from a mobile device. If you don't give a URL, the user is sent to the app's default start page after authentication completes. If the connected app that you're creating is a canvas app, then you can leave this field blank. The Canvas App URL field contains the URL that gets called for the connected app.
`oauthConfig`	connectedAppOauthConfig	Specifies how your app communicates with Salesforce.
`oauthPolicy`	ConnectedAppOauthPolicy	Specifies Oauth access policies associated with your connected app. Available in API version 49.0 and later.
`permissionSetName`	string[]	Specifies the permissions required to perform different functions with the connected app. Available in API version 46.0 and later. You can assign multiple permission sets to the connected app, but you must enter each permission set name on a separate line. You can't enter the same permission set name more than one time for each connected app. You can also change a permission set by replacing the current permission set with a new permission set. Make sure that each permission set name assigned to the connected app is unique. You can delete individual permission sets or remove all permission sets from a connected app by entering an empty `permissionSetName` string on deployment of the connected app: (`<permissionSetName></permissionSetName>`).

Field Name	Field Type	Description
		To use this field, the `isAdminApproved` field on the ConnectedAppOauthConfig subtype must be set to `true`.
`plugin`	string	The name of a custom Apex class that extends `Auth.ConnectedAppPlugin` to customize the behavior of the app.
`pluginExecutionUser`	string	Specifies the user to run the plugin as. If the user isn't authorized to use the connected app, use the `authorize` method. See the `ConnectedAppPlugin` class in the Apex Developer Guide. Available in API version 46.0 and later. Enter a user that is part of your org. Otherwise, the user is removed from this field when you deploy the connected app. If you don't want to specify a user, you can leave this field empty. To use this field in an org, the ConAppPluginExecuteAsUser setting must be enabled.
`profileName`	string[]	Specifies the profile (base-level user permissions) required to perform different functions with the connected app. Available in API version 46.0 and later. You can assign multiple profiles to the connected app, but you must enter each profile name on a separate line. You can't enter the same profile name more than one time for each connected app. You can also change profiles by replacing the current profiles with new profiles. Make sure that each profile name assigned to the connected app is unique. You can also delete individual profiles or remove all profiles from a connected app by entering an empty `profileName` string on deployment of the connected app: (`<profileName></profileName>`). To use this field, the `isAdminApproved` field on the ConnectedAppOauthConfig subtype must be set to `true`.
`samlConfig`	ConnectedAppSamlConfig	Controls how the app uses single sign-on.
`sessionPolicy`	ConnectedAppSessionPolicy	Specifies a connected app's session policies. Available in API version 49.0 and later.
`startUrl`	string	If the app isn't accessed from a mobile device, users are directed to this URL after they've authenticated. If you don't give a URL, the user is sent to the app's default start page after authentication completes. Whether you give a URL or not, the start URL can be updated later by managing the connected app. If the app is accessed from a mobile device, see `mobileStartUrl`. If the connected app that you're creating is a canvas app, then you can leave this field empty. The Canvas App URL field contains the URL that gets called for the connected app.

ConnectedAppAttribute

Represents the field names that make up a custom attribute when using SAML with a ConnectedApp. Tailor these values to a specific service provider.

Field Name	Field Type	Description
`formula`	string	Required. The value of the attribute.
`key`	string	Required. The attribute's identifier.

ConnectedAppCanvasConfig

Represents the configuration options of the connected app if it's exposed as a canvas app.

Field Name	Field Type	Description
`accessMethod`	AccessMethod (enumeration of type string)	Required. Indicates how the canvas app initiates the OAuth authentication flow. The valid values are: • `Get`—OAuth authentication is used, and the user is prompted to allow the third-party application to access their information. When you use this access method, the canvas app must initiate the OAuth authentication flow. • `Post`—OAuth authentication is used, but when the administrator installs the canvas app, they implicitly allow access for users. Therefore, the user isn't prompted to allow the third party to access their user information. When you use this access method, the authentication is posted directly to the canvas app URL.
`canvasUrl`	string	Required. The URL of the third-party app that's exposed as a canvas app.
`lifecycleClass`	string	The name of the `Canvas.CanvasLifecycleHandler` Apex class, if you've implemented this class for custom parameters. Available in API version 31.0 and later.
`locations`	CanvasLocationOptions (enumeration of type string)[]	Indicates where the canvas app can appear to the user. The valid values are: • `Aura`—Reserved for future use. • `AppLauncher`—Reserved for future use. • `Chatter`—The canvas app can appear in the app navigation list on the Chatter tab. • `ChatterFeed`—The canvas app can appear as a Chatter feed item. • `MobileNav`—The canvas app can appear in a mobile card in the Salesforce mobile app. Available in API version 31.0 and later. • `None`—The canvas app can appear only in the Canvas App Previewer.

Field Name	Field Type	Description
		• `OpenCTI`—The canvas app can appear in the call control tool. • `PageLayout`—The canvas app can appear on a page layout. When viewed in the Salesforce mobile app, the canvas app appears in the record detail page. Available in API version 31.0 and later. • `Publisher`—The canvas app can appear as a global action. • `ServiceDesk`—The canvas app can appear in the footer or sidebars of a Salesforce console. • `UserProfile`—Reserved for future use. • `Visualforce`—The canvas app can appear on a Visualforce page.
`options`	CanvasOptions (enumeration of type string)[]	Indicates whether to hide the **Share** button and header in the publisher for your canvas app and whether the app is a canvas personal app. Valid values are: • `HideShare`—The **Share** button is hidden in the publisher for the related canvas app. Available in API version 30.0 and later. • `HideHeader`—The header is hidden in the publisher for the related canvas app. Available in API version 30.0 and later. • `PersonalEnabled`—End users can install the app as a canvas personal app. Available in API version 32.0 and later.
`samlInitiationMethod`	SamlInitiationMethod (enumeration of type string)	If you're using SAML single sign-on (SSO), indicates which provider initiates the SSO flow. • `IdpInitiated`—Identity provider initiated. Salesforce makes the initial request to start the SSO flow. • `SpInitiated`—Service provider initiated. The canvas app starts the SSO flow after it's invoked. • `None`—The canvas app isn't using SAML SSO. Available in API version 31.0 and later.

ConnectedAppIpRange

Represents the list of IP addresses that can access the app without requiring the user to authenticate.

Field Name	Field Type	Description
`description`	string	Identifies the purpose of the range, such as which part of a network corresponds to this range. Available in API version 31.0 and later.
`end`	string	Required. The last address in the IP range, inclusive.
`start`	string	Required. The first address in the IP range, inclusive.

ConnectedAppOauthConfig

Represents the field names that configure how your connected app communicates with Salesforce.

Field Name	Field Type	Description
`assetTokenConfig`	connectedAppOauthAssetToken	The OAuth asset token configuration for the connected app OAuth settings. Available in API version 49.0 and later.
`callbackUrl`	string	Required. The endpoint that Salesforce calls back to your connected app during OAuth. It's the OAuth `redirect_uri`.
`certificate`	string	The PEM-encoded certificate string, if the app uses a certificate.
`consumerKey`	string	A value used by the consumer for identification to Salesforce. Referred to as `client_id` in OAuth 2.0. In API version 32.0 and later, you can set this field's value only during creation. After you define and save the value, it can't be edited. The value must be alphanumeric, can't contain special characters or spaces, and must be between 8–256 characters. Consumer keys must be globally unique.
`consumerSecret`	string	A value that is combined with the `consumerKey` and used by the consumer for identification to Salesforce. Referred to as `client_secret` in OAuth 2.0. Typically, Salesforce generates this value when you create the connected app. However, you can customize the shared secret value during creation. After you save the value, it can't be edited. When set, the value isn't returned in Metadata API requests. The value must be alphanumeric (no special characters and no spaces) and a minimum of 8 characters (maximum of 256 characters). If you specify a secret already in use for another connected app in the organization, an error occurs. Available in API version 32.0 and later.
`idTokenConfig`	ConnectedAppOauthIdToken	Specifies the ID token configuration for the connected app OAuth settings. Available in API version 43.0 and later.
`isAdminApproved`	boolean	If set to `false` (default), anyone in the org can authorize the app. Users must approve the app the first time they access it. If set to `true`, only users with the appropriate profile or permission set can access the app. These users don't have to approve the app before they can access it. Manage profiles for the app by editing each profile's Connected App Access list. Manage permission sets for the app by editing each permission set's Assigned Connected App list. This setting isn't available in Group Edition. Available in API version 46.0 and later. Connected app consumers can edit this setting when deploying a connected app in their org.

Field Name	Field Type	Description
isClientCredentialEnabled	boolean	If set to `true`, the connected app can use the OAuth 2.0 client credentials flow. To use the client credentials flow, you must also specify a user for `oauthClientCredentialUser`. If set to `false` (default), the connected app can't use the client credentials flow. Available in API version 56.0 and later.
isCodeCredentialEnabled	boolean	Determines whether the app can use the Authorization Code and Credentials Flow to provide identity services to headless, off-platform apps. The Authorization Code and Credentials Flow is the foundation of headless login, headless registration, headless passwordless login, and headless guest identity. If set to `true`, the connected app can use the Authorization Code and Credentials Flow and all associated Headless Identity features. The default value is `false`. This field is available in API version 57.0 and later.
isCodeCredentialPostOnly	boolean	For the Authorization Code and Credentials Flow, determines whether the user's credentials must be sent in the body of the initial HTTPS POST request to the Salesforce authorization endpoint. Requiring the credentials in the POST body instead of in the header improves security. If set to `true`, the user's credentials must be included in the POST body. The default value is `false`. This field is available in API version 57.0 and later.
isConsumerSecretOptional	boolean	If set to `false` (default), the connected app's client secret is required in exchange for an access token in the OAuth 2.0 web server flow. If the client app can't keep the client secret confidential and it must use the web server flow, set to `true`. A client secret is still generated for the connected app, but this setting instructs the web server flow not to require the `client_secret` parameter in the access token request. We recommend the user agent flow as a more secure option than web server flow without the secret. Available in API version 49.0 and later.
isIntrospectAllTokens	boolean	If set to `true`, authorizes the connected app to introspect all access and refresh tokens within the entire org. If set to `false` (default), the connected app can introspect its own tokens. In addition, an OAuth client that directly registers OAuth 2.0 connected apps through the dynamic client registration endpoint can check the tokens for itself and its registered apps. Available in API version 49.0 and later.

Field Name	Field Type	Description
isNamedUserJwtEnabled	boolean	If set to `true`, the connected app is enabled to issue JSON Web Token (JWT)-based access tokens. This field is generally available in API version 59.0 and later.
isPkceRequired	boolean	Determines whether the Proof Key for Code Exchange (PKCE) extension is required for variations of the OAuth 2.0 authorization code flow configured for the connected app, including the web server flow and Authorization Code and Credentials Flow. For public client apps that can't keep the consumer secret confidential, such as mobile apps, the PKCE extension helps ensure that the client that initiates an authorization flow is the same client that completes it. For this reason, we always recommend implementing PKCE for public clients. We also strongly recommend that you implement PKCE for private clients. If set to `true`, the PKCE extension is required and any authorization code flow variations that don't implement it fail. If set to `false`, you can still implement PKCE but it isn't required. The default value is `false`. This field is available in API version 59.0 and later.
isRefreshTokenRotationEnabled	boolean	If set to `true`, the connected app issues a new refresh token each time the OAuth refresh token flow is invoked. The old refresh token is automatically invalidated. If a user tries to use a previous refresh token that's been invalidated, the current refresh token and its associated access tokens get deleted. If set to `false`, the refresh token can be used to obtain multiple access tokens. This field is available in API version 60.0 and later.
isSecretRequiredForRefreshToken	boolean	If set to `true` (default), the app's client secret is required in the authorization request of a refresh token and hybrid refresh token flow. If set to `false` and an app sends the client secret in the authorization request, Salesforce still validates it. Select this option for web-server based apps that can protect client secrets. For apps that can't protect client secrets, such as mobile apps or apps installed on a user's computer, we recommend against selecting this option. Available in API version 51.0 and later.
isSecretRequiredForTokenExchange	boolean	If set to `true`, the connected app must include its consumer secret (`client_secret`) in the token request during the OAuth 2.0 token exchange flow. For security, set this field to `true` only if your app has a private client backend where it can keep the secret safe. For public client apps, such as single-page apps and mobile apps, set this field to `false` and don't include the consumer secret. This field is available in API version 60.0 and later.

Field Name	Field Type	Description
`isTokenExchangeEnabled`	boolean	If set to `true`, the connected app can use the OAuth 2.0 token exchange flow to exchange tokens from an external identity provider for Salesforce tokens. This field is available in API version 60.0 and later.
`oauthClientCredentialUser`	string	The execution user for the OAuth 2.0 client credentials flow. Salesforce returns access tokens on behalf of this user. This user must have the API Only permission. To use this field, set `isClientCredentialEnabled` to `true` and specify a `consumerKey`. Available in API version 56.0 and later.
`scopes`	ConnectedAppOauthAccessScope (enumeration of type string)[]	The permissions given by the user running the connected app. When deploying metadata, valid values are: • `Basic`—Allows access to your identity URL service (the same behavior as deploying `Address`, `Email`, `Phone`, and `Profile`). • `Api`—Allows access to the logged-in user's account over the APIs. • `Web`—Allows use of the `access_token` on the web. This usage also includes `visualforce`, allowing access to Visualforce pages. • `Full`—Allows access to all data accessible by the logged-in user. • `Chatter`—Allows access to only the Connect REST API resources. • `CustomApplications`—Provides access to custom applications, such as those using Visualforce. • `RefreshToken`—Allows a refresh token to be returned if you're eligible to receive one (the same behavior as deploying `OfflineAccess`). • `OpenID`—Allows access to the logged-in user's unique identifier for OpenID Connect apps. • `Profile`—Allows access to the logged-in user's profile (the same behavior as deploying `Basic`). • `Email`—Allows access to the logged-in user's email address (the same behavior as deploying `Basic`). • `Address`—Allows access to the logged-in user's street address (the same behavior as deploying `Basic`). • `Phone`—Allows access to the logged-in user's phone number value (the same behavior as deploying `Basic`). • `OfflineAccess`—Allows the app to interact with the user's data while the user is offline and get a refresh token (the same behavior as deploying `RefreshToken`).

Field Name	Field Type	Description
		• `CustomPermissions`—Allows access to the custom permissions in an organization associated with the connected app and shows whether the current user has each permission enabled. • `Wave`—Allows access to the Analytics REST API resources. Available in API version 35.0 and later. • `Eclair`—Allows access to the Analytics REST API Charts Geodata resource. Available in API version 35.0 and later. • `Pardot`—Allows access to Pardot API services on behalf of the user. The full extent of accessible services is managed by the Pardot account. Available in API version 49.0 and later. • `Lightning`—Allows hybrid apps to directly obtain Lightning child sessions through the OAuth 2.0 hybrid app token flow and hybrid app refresh token flow. Available in API version 51.0 and later. • `Content`—Allows hybrid apps to directly obtain content child sessions through the OAuth 2.0 hybrid app token flow and hybrid app refresh token flow. Available in API version 51.0 and later. • `CDPIngest`—Allows access to Data Cloud ingest API services. Customers use these API services to upload and maintain external datasets in the Data Cloud. Available in API version 52.0 and later. • `Chatbot`—Allows access to Einstein Bot API services. Available in API version 54.0 and later. • `ForgotPassword`—Allows access to Headless Forgot Password API. Assign to an internal integration user to get an access token for authenticated requests to this API. Available in API version 57.0 and later. • `UserRegistration`—Allows access to Headless Registration API. Assign to an internal integration user to get an access token for authenticated requests to this API. Available in API version 58.0 and later. • `PwdlessLogin`—Allows access to Headless Passwordless Login API. Assign to an internal integration user to get an access token for authenticated requests to this API. Available in API version 59.0 and later. When retrieving metadata, valid values are: • `Api`—Allows access to the logged-in user's account over the APIs. • `Basic`—Allows access to the user's identity URL service, and includes `Address`, `Email`, `Phone`, and `Profile`. • `Chatter`—Allows access to only the Connect REST API resources. • `CustomApplications`—Allows access to custom applications, such as those using Visualforce. • `Full`—Allows access to all data accessible by the logged-in user.

Field Name	Field Type	Description
		• `OpenID`—Allows access to the logged-in user's unique identifier for OpenID Connect apps. • `CDPIngest`—Allows access to Data Cloud ingest API services. Customers use these API services to upload and maintain external datasets in the Data Cloud. Available in API version 52.0 and later. • `Pardot`—Allows access to Pardot API services on behalf of the user. The full extent of accessible services is managed by the Pardot account. Available in API version 49.0 and later. • `Lightning`—Allows hybrid apps to directly obtain Lightning child sessions through the OAuth 2.0 hybrid app token flow and hybrid app refresh token flow. Available in API version 51.0 and later. • `Content`—Allows hybrid apps to directly obtain content child sessions through the OAuth 2.0 hybrid app token flow and hybrid app refresh token flow. Available in API version 51.0 and later. • `RefreshToken`—Allows a refresh token to be returned if you're eligible to receive one and is synonymous with allowing `OfflineAccess`. • `Wave`—Allows access to the Analytics REST API resources. Available in API version 35.0 and later. • `Eclair`—Allows access to the Analytics REST API Charts Geodata resource. Available in API version 35.0 and later. • `Web`—Allows usage of the `access_token` on the web. This usage also includes `visualforce`, allowing access to Visualforce pages. • `Chatbot`—Allows access to Einstein Bot API services. Available in API version 54.0 and later. • `ForgotPassword`—Allows access to Headless Forgot Password API. Assign to an internal integration user to get an access token for authenticated requests to this API. Available in API version 57.0 and later. • `UserRegistration`—Allows access to Headless Registration API. Assign to an internal integration user to get an access token for authenticated requests to this API. Available in API version 58.0 and later. • `PwdlessLogin`—Allows access to Headless Passwordless Login API. Assign to an internal integration user to get an access token for authenticated requests to this API. Available in API version 59.0 and later.
`singleLogoutUrl`	string	The single logout endpoint. This URL is the endpoint where Salesforce sends a logout request when users log out of Salesforce.

ConnectedAppOauthAssetToken

Specifies an OAuth asset token configuration for the connected app OAuth settings. Available in API version 49.0 and later.

Field Name	Field Type	Description
`assetAudiences`	string	Required. The audience claim associated with the asset token payload. This claim identifies who the JWT is intended for. Value is an array of case-sensitive strings, each containing a `StringOrURI` value. An audience is specified for each intended consumer of the asset token.
`assetIncludeAttributes`	boolean	Required. If set to `true` (default), custom attributes associated with the connected app are included in the asset token payload. If set to `false`, these attributes aren't included.
`assetIncludeCustomPerms`	boolean	Required. If set to `true` (default), custom permissions associated with the connected app are included in the asset token payload. If set to `false`, these permissions aren't included.
`assetSigningCertId`	string	Required. The ID of the JWT certificate's signing secret. The certificate size can't exceed 4 KB. If it does, try using a DER encoded file to reduce the size.
`assetValidityPeriod`	int	Required. The asset token's validity period. The validity must be the expiration time of the assertion within 3 minutes, expressed as the number of seconds from 1970-01-01T0:0:0Z measured in UTC.

ConnectedAppOauthIdToken

Specifies the ID token configuration for the connected app OAuth settings. Available in API version 43.0 and later.

Field Name	Field Type	Description
`idTokenAudience`	string	The audiences that this ID token is intended for. The value is an array of case-sensitive strings. If no audiences are specified, the OAuth 2.0 `client_id` of the relying party is returned as the default audience. Otherwise, the other audiences are returned with the `client_id` in the `aud` value.
`idTokenIncludeAttributes`	boolean	Indicates whether attributes are included in the ID token.
`idTokenIncludeCustomPerms`	boolean	Indicates whether custom permissions are included in the ID token.
`idTokenIncludeStandardClaims`	boolean	Indicates whether standard claims about the authentication event are included in the ID token.
`idTokenValidity`	int	The length of time that the ID token is valid for after it's issued. The value can be from 1 to 720 minutes. The default is 2 minutes.

ConnectedAppOauthPolicy

Specifies OAuth access policies for the connected app. Available in API version 49.0 and later.

Field Name	Field Type	Description
`ipRelaxation`	string	Required. Specifies whether a user's access to the connected app is restricted by IP ranges. Valid options are: • `ENFORCE` (default)—Enforces the IP restrictions configured for the org, such as the IP ranges assigned to a user profile. • `BYPASS_2FACTOR`—Allows a user running the app to bypass the org's IP restrictions when either of these conditions is true. – The app has a list of allowed IP ranges and is using the web server OAuth authorization flow. Requests coming from only these IPs are allowed. – The app doesn't have a list of allowed IP ranges, but it uses the web server authentication flow. And the user successfully completes identity verification if accessing Salesforce from a new browser or device. • `BYPASS`—Allows a user to run this app without org IP restrictions. • `ENFORCE_RELAXREFRESH`—Enforces the IP restrictions configured for the org, such as the IP ranges assigned to a user profile. However, this option bypasses these restrictions when the connected app uses refresh tokens to get access tokens.
`refreshTokenPolicy`	string	Required. Specifies how long a refresh token is valid for. If refresh tokens are provided, users can continue to access the OAuth-enabled connected app without having to reauthorize when the access token expires, as defined by the session timeout value. The connected app exchanges the refresh token with an access token to start a new session. The Refresh Token policy is evaluated only during usage of the issued refresh token and doesn't affect a user's current session. Refresh tokens are required only when a user's session has expired or isn't available. For example, you set a refresh token policy to expire the token after 1 hour. If a user uses the app for 2 hours, the user isn't forced to reauthenticate after 1 hour. However, the user is required to authenticate again when the session expires and the client attempts to exchange its refresh token for a new session. Valid options are: • `zero`—The refresh token is invalid immediately. The user can use the current session (access token) already issued, but can't obtain a new session when the access token expires. • `infinite`—The refresh token is used indefinitely, unless revoked by the user or Salesforce admin. Default setting. • `specific_lifetime:`***`number`***`:`***`HOURS, DAYS, MONTHS`***—The refresh token is valid for a fixed amount of time. For example, if the policy states `specific_lifetime:1:DAYS`, the user can obtain new sessions for only 24 hours.

Field Name	Field Type	Description
		• `specific_inactivity:`***`number`***`:`***`HOURS, DAYS, MONTHS`***—The refresh token is valid as long as it's been used within the specified amount of time. For example, if set to `specific_inactivity:7:DAYS`, and the refresh token isn't exchanged for a new session within seven days, the next attempt to use the token fails. The expired token can't generate new sessions. If the refresh token is exchanged within seven days, the token is valid for another seven days. The monitoring period of inactivity also resets.
`singleLogoutUrl`	string	If single logout is enabled, specify the single logout URL. Salesforce sends logout requests to this URL when users log out of Salesforce. The single logout URL must be an absolute URL starting with `https://`.

ConnectedAppSamlConfig

Specifies how an app uses single sign-on.

Field Name	Field Type	Description
`acsUrl`	string	Required. The assertion consumer service URL from the service provider.
`certificate`	string	The PEM-encoded certificate string, if the app uses a certificate.
`encryptionCertificate`	string	The name of the certificate to use for encrypting SAML assertions to the service provider. This certificate is saved in the organization's Certificate and Key Management list. Available in API version 30.0 and later.
`encryptionType`	SamlEncryptionType (enumeration of type string)	When Salesforce is the identity provider, the SAML configuration can specify the encryption method used for encrypting SAML assertions to the service provider. The service provider detects the encryption method in the SAML assertion for decryption. Valid values are: • `AES_128`—128–bit key • `AES_256`—256–bit key Available in API version 30.0 and later.
`entityUrl`	string	Required. The entity ID from your service provider.
`issuer`	string	A URI that sends the SAML response. A service provider can use this URI to determine which identity provider sent the response. Available in API version 29.0 and later.
`samlIdpSLOBindingEnum`	SamlIdpSLOBinding (enumeration of type string)	The SAML HTTP binding type from the service provider used for single logout. Available in API version 40.0 and later. Valid values are: • `PostBinding` • `RedirectBinding`

Field Name	Field Type	Description
samlNameIdFormat	SamlNameIdFormatType (enumeration of type string)	Indicates the format the service provider (SP) requires for the user's single sign-on identifier. Available in API version 29.0 and later. Valid values are: • Unspecified (default)—No format given. • EmailAddress—Used if the subject type is the user's name or a federation ID (an ID internal to the SP). • Persistent—Used with the user ID and persistent ID subject types. • Transient—Used when the subject type is a custom attribute and can change every time the user logs in.
samlSigningAlgoType	SamlSigningAlgoType (enumeration of type string)	Indicates the signing algorithm applied to SAML requests and responses when Salesforce is the identity provider. The selected signing algorithm is applied to both single sign-on and single logout responses from your org. Available in API version 50.0 and later. Valid values are: • SHA1 • SHA256
samlSloUrl	string	The SAML single-logout endpoint of the connected app service provider (SP). This endpoint is where SAML LogoutRequests and LogoutResponses are sent when users log out of Salesforce. The SP provides this endpoint. Available in API version 40.0 and later.
samlSubjectCustomAttr	string	If the samlSubjectType is CustomAttr, include that custom value here; otherwise, leave empty. Available in API version 29.0 and later.
samlSubjectType	SamlSubjectType (enumeration of type string)	Required. The single sign-on identifier for the user. Valid values are: • Username—The user's Salesforce name. • FederationId—The user's identifier at the service provider. Get this value from the service provider. • UserId—The user's 15-character Salesforce identifier. • PersistentID—A persistent opaque identifier that is specific to the identity provider and a service provider. • CustomAttr—The identifier is taken from a custom field value in samlSubjectCustomAttr.

ConnectedAppSessionPolicy

Specifies the configuration options for a connected app's session policies. Use these policies to define how long a user's session can last before reauthenticating, to block user access to the connected app, or to require multi-factor authentication (MFA) to access the app. Available in API version 49.0 and later.

Field Name	Field Type	Description
`policyAction`	string	If the High Assurance session security level is applied to the connected app, specify associated high assurance action. Valid values are: • `Block`—Makes the connected app inaccessible to your org's users. Blocking an app ends all current user sessions with the connected app and prevents all new sessions. • `RaiseSessionLevel`—Requires users to verify their identity with multi-factor authentication when they log in to the connected app. This setting applies to authorization flows that include a user approval step for API logins. These flows are the OAuth 2.0 refresh token flow, web server flow, and user-agent flow. All other flows, such as the JSON Web Token (JWT) bearer token flow, don't include a user approval step. For flows without a user approval step, API logins with the High Assurance session security level are blocked.
`sessionLevel`	string	Applies the High Assurance session security level to the connected app. This session level requires users to verify their identity with multi-factor authentication when they log in to the connected app.
`sessionTimeout`	int	The length of time the connected app's session lasts. If you don't set a value, Salesforce uses the timeout value in the connected app user's profile. If the user's profile doesn't specify a timeout value, Salesforce uses the timeout value in the org's Session Settings.

Declarative Metadata Sample Definition

The following is an example of a ConnectedApp component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ConnectedApp xmlns="http://soap.sforce.com/2006/04/metadata">
    <attributes>
        <formula>$Api.Enterprise_Server_URL_100</formula>
        <key>test</key>
    </attributes>
    <attributes>
        <formula>$Api.Partner_Server_URL_60</formula>
        <key>test1</key>
    </attributes>
   <canvasConfig>
        <accessMethod>Get</accessMethod>
        <canvasUrl>https://salesforce.com</canvasUrl>
        <lifecycleClass>MyCanvasListener</lifecycleClass>
        <locations>Chatter</locations>
        <locations>Visualforce</locations>
        <locations>Aura</locations>
        <locations>Publisher</locations>
        <locations>ChatterFeed</locations>
        <locations>OpenCTI</locations>
        <locations>MobileNav</locations>
        <locations>PageLayout</locations>
```

```
        <options>HideShare</options>
        <options>HideHeader</options>
        <options>PersonalEnabled</options>
        <samlInitiationMethod>None</samlInitiationMethod>
    </canvasConfig>
    <canvas>
     <locationOptions>NONE</locationOptions>
     <samlInitiationMethod>None</samlInitiationMethod>
     <accessMethod>Get</accessMethod>
     <canvasOptions>PE</canvasOptions>
     <lifecycleClass>MyCanvasListener</lifecycleClass>
     <canvasUrl>https://salesforce.com</canvasUrl>
 </canvas>
    <contactEmail>example@salesforce.com</contactEmail>
    <contactPhone>1231231234</contactPhone>
    <description>Test App</description>

<iconUrl>https://c1.sfdcstatic.com/content/dam/sfdc-docs/www/logos/salesforce-logo-cloud.png</iconUrl>


<infoUrl>https://c1.sfdcstatic.com/content/dam/sfdc-docs/www/logos/salesforce-logo-cloud.png</infoUrl>

    <startUrl>https://www.salesforce.com</startUrl>
    <ipRanges>
        <end>000.0.0.1</end>
        <start>000.0.0.2</start>
 <description>Test</description>
    </ipRanges>
    <ipRanges>
        <end>000.0.0.1</end>
        <start>000.0.0.2</start>
 <description>Test1</description>
    </ipRanges>
    <label>TestApp</label>

<logoUrl>https://c1.sfdcstatic.com/content/dam/sfdc-docs/www/logos/salesforce-logo-cloud.png</logoUrl>

    <profileName>Test</profileName>
    <permissionSetName>TestPermission</permissionSetName>
    <mobileStartUrl>http://www.mobile.com</mobileStartUrl>
    <mobileAppConfig>
        <applicationBinaryFile></applicationBinaryFile>
        <applicationBinaryFileName>test</applicationBinaryFileName>
        <applicationBundleIdentifier>testtest</applicationBundleIdentifier>
        <applicationIconFileName>test</applicationIconFileName>
 <applicationIconFile>test</applicationIconFile>
 <applicationFileLength>5</applicationFileLength>
        <applicationInstallUrl>https://salesforce.com</applicationInstallUrl>
        <devicePlatform>ios</devicePlatform>
        <deviceType>minitablet</deviceType>
        <minimumOsVersion>2</minimumOsVersion>
        <privateApp>true</privateApp>
        <version>2</version>
    </mobileAppConfig>
```

```
    <oauthConfig>
        <assetTokenConfig>
            <assetAudiences>http://asset.audience.com</assetAudiences>
            <assetIncludeAttributes>true</assetIncludeAttributes>
            <assetIncludeCustomPerms>true</assetIncludeCustomPerms>
            <assetSigningCertId>${cert.id}</assetSigningCertId>
            <assetValidityPeriod>1440</assetValidityPeriod>
        </assetTokenConfig>
        <callbackUrl>https://www.callback.com</callbackUrl>
        <!--  NOTE, TEST.orgId will get replaced with the org ID of the context org, so
we will have a unique consumer key in every scratch org.  -->
        <consumerKey>3MVG9AOp4kbriZOcnmoLmTrguy9ryzcLbBjoNY...${TEST.orgId}</consumerKey>

  <consumerSecret>3MVG9AOp4k...</consumerSecret>
  <certificate>3MVG9AOp4kbriZOInmoLmTrguy9ryzcLbBjoNY...</certificate>
        <scopes>Basic</scopes>
        <scopes>Chatter</scopes>
        <scopes>OpenID</scopes>
        <scopes>CustomPermissions</scopes>
  <singleLogoutUrl>https://www.logout.com</singleLogoutUrl>
        <isAdminApproved>false</isAdminApproved>
        <isConsumerSecretOptional>false</isConsumerSecretOptional>
        <isIntrospectAllTokens>false</isIntrospectAllTokens>
  <idTokenConfig>
   <idTokenAudience>https://idtoken.audience.com</idTokenAudience>
   <idTokenIncludeAttributes>true</idTokenIncludeAttributes>
   <idTokenIncludeCustomPerms>true</idTokenIncludeCustomPerms>
   <idTokenIncludeStandardClaims>true</idTokenIncludeStandardClaims>
   <idTokenValidity>20</idTokenValidity>
  </idTokenConfig>
    </oauthConfig>
    <oauthPolicy>
        <ipRelaxation>ENFORCE</ipRelaxation>
        <refreshTokenPolicy>infinite</refreshTokenPolicy>
        <singleLogoutUrl>https://www.logout.com</singleLogoutUrl>
    </oauthPolicy>
    <plugin>ConnectedAppPluginTest</plugin>
    <pluginExecutionUser>testuser@salesforce.com</pluginExecutionUser>
    <samlConfig>
        <acsUrl>http://www.acs.com</acsUrl>
        <encryptionType>AES_128</encryptionType>
 <encryptionCertificate>3MVG9AOp4kbriZOInmoLmTrguy9ryzcLbBjoNY...</encryptionCertificate>

 <certificate>3MVG9AOp4kbriZOInmoLmTrguy9ryzcLbBjoNY...</certificate>
 <samlSubjectCustomAttr>test</samlSubjectCustomAttr>
        <entityUrl>http://www.entity.com</entityUrl>
        <issuer>https://salesforce.com</issuer>
        <samlIdpSLOBindingEnum>RedirectBinding</samlIdpSLOBindingEnum>
        <samlNameIdFormat>Unspecified</samlNameIdFormat>
        <samlSloUrl>https://www.salesforce.com</samlSloUrl>
        <samlSubjectType>CustomAttribute</samlSubjectType>
    </samlConfig>
    <sessionPolicy>
        <policyAction>RaiseSessionLevel</policyAction>
```

```
        <sessionLevel>HIGH_ASSURANCE</sessionLevel>
        <sessionTimeout>720</sessionTimeout>
    </sessionPolicy>
</ConnectedApp>
```

You can enter multiple callback URL values. At run time, Salesforce validates the callback URL specified by the app by matching it with one of the values. You must separate each callback URL with line breaks. To enter a new line programmatically, use the `\r` line break character.

Here's an example of a ConnectedApp component with multiple callback URLs.

```
<?xml version="1.0" encoding="UTF-8"?>
<ConnectedApp xmlns="http://soap.sforce.com/2006/04/metadata">
 <contactEmail>example@salesforce.com</contactEmail>
 <label>MyConnectedApp</label>
 <oauthConfig>
 <callbackUrl>https://example.com/callback1
https://example.com/callback2
https://example.com/callback3</callbackUrl>
 <consumerKey>3MVG9AOp4kbriZOcnmoLmTrguy9ryzcLbBjoNY...</consumerKey>
 <isAdminApproved>false</isAdminApproved>
 <isConsumerSecretOptional>false</isConsumerSecretOptional>
 <isIntrospectAllTokens>false</isIntrospectAllTokens>
 <isSecretRequiredForRefreshToken>true</isSecretRequiredForRefreshToken>
 <scopes>Full</scopes>
 <scopes>RefreshToken</scopes>
 </oauthConfig>
 <oauthPolicy>
 <ipRelaxation>ENFORCE</ipRelaxation>
 <refreshTokenPolicy>infinite</refreshTokenPolicy>
 </oauthPolicy>
</ConnectedApp>
```

The following is an example package manifest used to deploy or retrieve the ConnectedApp metadata for an organization.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>PortalTestApp</members>
        <name>ConnectedApp</name>
    </types>
    <version>29.0</version>
</Package>
```

Usage

If you're constructing a SAML-enabled connected app using Metadata API, and must set the `IdP-Initiated Login URL` for your service provider, you have two options:

You can use the service provider app ID with the `app` parameter in the following format. This value is displayed in the Salesforce user interface. From Setup, enter *`Connected Apps`* in the Quick Find box, then select **Connected Apps**, then click the name of the connected app to see its detail page.

```
https://<Salesforce_base_URL>/idp/login?app=<app_id>
```

Or, if you're configuring the connected app using Metadata API only, you can use the `apiName` parameter of the service provider app in the following format. The `apiName` parameter is the `fullName` inherited from the Metadata type.

```
https://<Salesforce_base_URL>/idp/login?apiName=<fullName>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ContentAsset

Represents the metadata for creating an asset file. Asset files enable a Salesforce file to be used for org setup and configuration purposes. This type extends the MetadataWithContent metadata type and inherits its `content` and `fullName` fields.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

ContentAsset components have the suffix `.asset` and are stored in the `contentassets` folder.

Version

ContentAsset components are available in API version 38.0 and later.

Special Access Rules

The system prevents metadata retrieval if the total size of the asset's file content exceeds 30 MB. All pre-existing limits for packaging apply to asset files.

Fields

Field Name	Field Type	Description
`format`	ContentAssetFormat (enumeration of type string)	Describes the format of the asset file. Valid values are: • `Original`—A single asset file version. • `ZippedVersions`—Contains multiple versions of the asset file.
`isVisibleByExternalUsers`	boolean	Indicates whether unauthenticated users can see the asset file (`true`) or not (`false`). If not specified, the default value is `false`. This field is available in API version 44.0 and later.
`language`	string	Required. The language of the asset file label.
`masterLabel`	string	Required. The label for the asset file record, which displays in Setup.

Field Name	Field Type	Description
originNetwork	string	For deploys, the name of the Experience Cloud site the file is assigned upon creation. For retrievals, the name of the Experience Cloud site the file is assigned to appears in the field value. If null, the file wasn't assigned to an Experience Cloud site.
relationships	ContentAssetRelationships	The list of ContentAssetLinks that describe whether the asset file can be shared with the org.
versions	ContentAssetVersions	Required. Captures basic information about the file version included the asset metadata. Typically the file has only one version.

ContentAssetRelationships

Represents the relationships between an asset file and the locations it's linked with.

Field Name	Field Type	Description
emailTemplate	ContentAsset[]	An array of email templates the content asset is related to. This field is available in API version 51.0 and later.
insightsApplication	ContentAsset[]	An array of the insights applications that use the content asset. This field is available in API version 39.0 and later.
network	ContentAsset[]	An array of networks that use the content asset. This field is available in API version 39.0 and later.
organization	ContentAsset[]	Stores information about sharing the asset file with the org. Maps to ContentDocumentLink. This field is available in API version 39.0 and later.
workspace	ContentAsset[]	An array of workspaces and libraries that own or share the content asset. This field is available in API version 39.0 and later.

ContentAssetLink

Represents a relationship link for an asset file, and includes details about the level of access for the link.

Field Name	Field Type	Description
access	ContentAssetAccess (enumeration of type string)	Required. The permission granted to the user of the shared file, determined by the permission the user already has. Valid values are: • VIEWER • COLLABORATOR • INFERRED
isManagingWorkspace	boolean	Indicates whether the content asset resides in the workspace or not. When true, the content asset resides in the workspace. If not specified, the default value is false. This field is available in API version 39.0 and later.

Field Name	Field Type	Description
name	string	Reserved for future use.

ContentAssetVersions

Represents information about all file versions included in the asset metadata.

Field Name	Field Type	Description
version	ContentAssetVersion[]	A list of file versions for the asset.

ContentAssetVersion

Represents information about one file version included in the asset metadata.

Field Name	Field Type	Description
number	string	Required. The version number. This field is based on, or sets, the ContentVersion.
pathOnClient	string	Required. Describes the original filename of the file. This field maps to ContentVersion.PathOnClient. It provides the data for the ContentVersion Title field.
zipEntry	string	If the asset file has more than one version, `format` is `ZippedVersions`. In this case, `zipEntry` is the name of the file within the zip. If the asset file has only one version, this field is empty.

Declarative Metadata Sample Definition

The following is an example of a ContentAsset component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ContentAsset xmlns="http://soap.sforce.com/2006/04/metadata">
    <masterLabel>some asset</masterLabel>
    <relationships>
        <organization>
            <access>VIEWER</access>
        </organization>
    </relationships>
    <versions>
        <version>
            <number>1</number>
            <pathOnClient>some asset.txt</pathOnClient>
        </version>
    </versions>
</ContentAsset>
```

For assets that include just one version, the format field can be omitted or specified with the value as `Original`. File assets with more than one version have versions wrapped in a zip file.

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>MyAsset</members>
        <name>ContentAsset</name>
    </types>
    <version>64.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ContentTypeBundle

Represents the definition of enhanced custom content types for use with enhanced CMS workspaces. When you create an enhanced custom content type, deploy this bundle to your org. Enhanced custom content types are displayed as forms with defined fields. When deployed, enhanced custom content types are available for use with enhanced LWR site channels. To use enhanced custom content types with Aura and non-enhanced LWR site channels, use enhanced CMS workspaces resources.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata type and inherits its `fullName` field.

ContentTypeBundle Structure and Directory Location

ContentTypeBundle components are stored in the `contentTypes` folder. Here's an example of how the folder is structured.

```
+--myMetadataPackage
    +--contentTypes (1)
        +--bbHost (2)
           +--schema.json (3)
```

- The contentTypes folder (1) contains a folder for each enhanced custom content type.
- Each enhanced custom content type folder is named in the format *`contentTypeName`*. In this example (2), the name is *`bbHost`*.
- Each contentTypeName folder contains a JSON file, `schema.json` (3), that defines the enhanced custom content type. The JSON file contains a title and one or more Lightning property types. Use this file to edit the properties of the enhanced custom content type on your local machine or scratch org and then deploy it.

Version

ContentTypeBundle components are available in version 64.0 and later.

Special Access Rules

ContentTypeBundle is available only when Salesforce CMS and digital experiences are enabled for your org.

Fields

Name	Description
`description`	**Type** string **Description** Explanatory text about the content type.
`masterLabel`	**Type** string **Description** Required. A name for ContentTypeBundle, which is defined when the ContentTypeBundle is created.
`resources`	**Type** ContentTypeBundleResource[] **Description** A list of source files in the ContentTypeBundle folder.

ContentTypeBundleResource

Represents the resource file inside the ContentTypeBundle.

Name	Description
`fileName`	**Type** string **Description** Required. The name of the resource file.
`filePath`	**Type** string **Description** Required. The path to the resource.

Name	Description
source	**Type** base64Binary **Description** Required. The content of the resource.

Declarative Metadata Sample Definition

This `package.xml` retrieves all of the ContentTypeBundle components in an org.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>ContentTypeBundle</name>
    </types>
    <version>64.0</version>
</Package>
```

In the retrieved `.zip` file, each enhanced custom content type is nested under a contentTypes folder.

This example shows the directory structure in the `.zip` file of an enhanced custom content type. The enhaced custom content type is named bbHost and in this example it represents the profile of a bed and breakfast host.

```
contentTypes
    bbHost
        schema.json
```

Here are the example contents of the `schema.json` file in the contentTypes directory. The bbHost is a complex type that includes subproperties for `fullName`, `aboutMe`, `interests`, `profilePicture`, `dateOfBirth`, `lastModified`, `personalWebsite`, `hostIdentityVerified`, `hostingExperienceInYears`, `rating`, `residenceCountry`, and `preferredModeOfContact`. Each subproperty is a primitive type. The subproperty types included in the `schema.json` file are completely configurable and must include a `lightning:type` for each property.

```
{
  "title": "Bed & Breakfast Host",
  "description": "Schema for capturing B&B host details",
  "lightning:type": "lightning__objectType",
  "lightning:mixinTypes": {
    "sfdc_cms:metadataContent": {}
  },
  "properties": {
    "fullName": {
      "title": "Full Name",
      "lightning:type": "lightning__textType",
      "lightning:textIndexed": true,
      "minLength": 5,
      "maxLength": 50,
      "lightning:uiOptions": {
```

```
      "placeholderText": "Enter your full name (e.g., John Doe)"
    },
    "lightning:localizable": false
  },
  "aboutMe": {
    "title": "About Me",
    "lightning:type": "lightning__richTextType",
    "minLength": 10,
    "maxLength": 3000,
    "lightning:textIndexed": false,
    "lightning:uiOptions": {
      "placeholderText": "Write something about yourself"
    },
    "lightning:localizable": true
  },
  "interests": {
    "title": "Interests and Hobbies",
    "lightning:type": "lightning__multilineTextType",
    "lightning:textIndexed": true,
    "minLength": 10,
    "maxLength": 500,
    "lightning:uiOptions": {
      "placeholderText": "Summarize your interests and hobbies in a few lines"
    },
    "lightning:localizable": true
  },
  "profilePicture": {
    "title": "Profile Picture",
    "lightning:type": "lightning__imageType",
    "lightning:uiOptions": {
      "placeholderText": "Upload a professional headshot"
    },
    "lightning:localizable": false
  },
  "dateOfBirth": {
    "title": "Date of Birth",
    "lightning:type": "lightning__dateType",
    "lightning:localizable": true,
    "lightning:uiOptions": {
      "placeholderText": "Select your date of birth"
    }
  },
  "lastModified": {
    "title": "Last Profile Update",
    "lightning:type": "lightning__dateTimeType",
    "lightning:localizable": false,
    "lightning:uiOptions": {
      "placeholderText": "Auto-filled on profile update"
    }
  },
  "personalWebsite": {
    "title": "Personal Website",
    "lightning:type": "lightning__urlType",
    "lightning:localizable": false,
```

```
      "lightning:uiOptions": {
        "placeholderText": "https://yourwebsite.com"
      }
    },
    "hostIdentityVerified": {
      "title": "Host Identity Verified",
      "lightning:type": "lightning__booleanType",
      "lightning:uiOptions": {
        "placeholderText": "Check if host identity is verified"
      }
    },
    "hostingExperienceInYears": {
      "title": "Years of Experience hosting B&B",
      "lightning:type": "lightning__integerType",
      "minimum": 0,
      "maximum": 50,
      "lightning:localizable": false,
      "lightning:uiOptions": {
        "placeholderText": "Enter total years of experience being a B&B host"
      }
    },
    "rating": {
      "title": "Rating",
      "lightning:type": "lightning__numberType",
      "minimum": 0.0,
      "maximum": 5.0,
      "lightning:localizable": false,
      "lightning:uiOptions": {
        "placeholderText": "e.g., 4.5"
      }
    },
    "residenceCountry": {
      "title": "Country of Residence",
      "lightning:type": "lightning__textType",
      "const": ["India"]
    },
    "preferredModeOfContact": {
      "title": "Preferred Mode of Contact",
      "lightning:type": "lightning__textType",
      "enum": ["email", "SMS", "phone"],
      "lightning:uiOptions": {
        "placeholderText": "Choose your preferred mode of contact"
      }
    }
  },
  "required": ["fullName", "hostIdentityVerified"]
```

Usage

For each custom content type that you create, you must also create a CMS content page created in the enhanced LWR, LWR, or Aura site that displays the content. Each content page serves as the detail page for all content of a single content type. See Create Custom Site Pages with Experience Builder.

To use enhanced custom content types with Aura and non-enhanced LWR site channels, use the enhanced CMS workspaces resources for CMS Delivery Contents and CMS Delivery Content.

Wildcard Support in the Manifest

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving with the Zip .

ContextDefinition

Represents the details of a context definition that describe the relationship between the node structures within a context.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

ContextDefinition components have the suffix `.contextDefinition` and are stored in the `contextDefinitions` folder.

Version

ContextDefinition components are available in API version 59.0 and later.

Special Access Rules

Enable the organization preference ContextDefinitionsEnabled to access the ContextDefinition metadata type.

Fields

Field Name	Description
`canBeReferenceDefinition`	**Field Type** boolean **Description** Indicates whether the context definition can be referred by other context definitions (`true`) or not (`false`). Available in API version 63.0 and later. The default value is `false`.
`clonedFrom`	**Field Type** string

Field Name	Description
	Description The name of the context definition that's used to clone the current context definition.
`contextDefinitionReferences`	**Field Type** ContextDefinitionReference[] **Description** References of the context definition.
`contextDefinitionVersions`	**Field Type** ContextDefinitionVersion[] **Description** Version of the context definition.
`contextTtl`	**Field Type** int **Description** Duration to persist the data, which is loaded in the run-time context instances created by this context definition, in the cache. The default value is 10 minutes.
`description`	**Field Type** string **Description** Description of the context definition.
`hasSystemTags`	**Field Type** boolean **Description** Indicates whether the context definition has system tags (`true`) or not (`false`). Available in API version 63.0 and later. The default value is `false`.
`inheritedFrom`	**Field Type** string **Description** Name of the parent context definition that's used to derive the current context definition.
`inheritedFromVersion`	**Field Type** string

Field Name	Description
	Description Version number of the parent definition that's used to derive the current context definition.
`isProtected`	**Field Type** boolean **Description** Auto-generated value that doesn't impact the behavior of the metadata type.
`masterLabel`	**Field Type** string **Description** Required. User-friendly name for the context definition, which is defined when the context definition is created.
`title`	**Field Type** string **Description** Required. Name of the context definition.

ContextDefinitionReference

Represents details about the context definition reference.

Field Name	Description
`inheritedFrom`	**Field Type** string **Description** ID of the parent context definition reference that's used to derive the current context definition reference.
`referenceContextDefinition`	**Field Type** string **Description** Required. ID or name of the referred context definition.

ContextDefinitionVersion

Represents details about the context definition version. Only one version can be active at a time.

Field Name	Description
`contextMappings`	**Field Type** ContextMapping[] **Description** Mapping of attributes and nodes to related objects.
`contextNodes`	**Field Type** ContextNode[] **Description** Details of the structure of the nodes within the context.
`endDate`	**Field Type** string **Description** Date and time when the context definition version becomes inactive.
`isActive`	**Field Type** boolean **Description** Indicates whether the context definition version is active (`true`) or not (`false`). The default value is `false`.
`startDate`	**Field Type** string **Description** Required. Date and time when the context definition version becomes active.
`versionNumber`	**Field Type** int **Description** Required. Version number of the context definition.

ContextMapping

Represents the mapping of attributes and nodes to related objects.

Field Name	Description
`contextMappingIntents`	**Field Type** ContextMappingIntent[] **Description** Purpose associated to a context mapping.
`contextNodeMappings`	**Field Type** ContextNodeMapping[] **Description** Mapping of the node in the context and values in the input schema.
`default`	**Field Type** boolean **Description** Indicates whether the mapping for a context definition version is default (`true`) or not (`false`). The default value is `false`.
`description`	**Field Type** string **Description** Description of the context mapping.
`inheritedFrom`	**Field Type** string **Description** Name of the parent mapping that's used to derive the current mapping.
`title`	**Field Type** string **Description** Required. Name of the context mapping.

ContextMappingIntent

Represents the purpose associated to a context mapping.

Field Name	Description
`mappingIntent`	**Field Type** ContextMappingIntentType (enumeration of type string)

Field Name	Description
	Description Required. Specifies the purpose that's used to identify the type of context mapping required. Valid values are: • `Hydration` • `Association` • `Persistence` • `Translation`

ContextNodeMapping

Represents the relationship between the node in the context and values in the input schema.

Field Name	Description
`contextAttributeMappings`	**Field Type** ContextAttributeMapping[] **Description** Mapping of the attribute defined in the context and the values in the related objects.
`contextNode`	**Field Type** string **Description** Context node record associated with the context node mapping.
`inheritedFrom`	**Field Type** string **Description** Name of the parent context node mapping that's used to derive the current context node mapping.
`mappedContextDefinition`	**Field Type** string **Description** API name of the context definition for existing context-to-context mappings.
`object`	**Field Type** string **Description** Name of the object used for the mapping.

ContextAttributeMapping

Represents the relationship between the attributes defined in the context and the values in the related objects.

Field Name	Description
`contextAttrHydrationDetails`	**Field Type** ContextAttrHydrationDetail[] **Description** Details of the SOQL (database) queries that fetch data for a chosen attribute from the input schema.
`contextAttribute`	**Field Type** string **Description** Context attribute record associated with the context attribute mapping.
`contextInputAttributeName`	**Field Type** string **Description** Required. Name of the input attribute.
`ctxAttrHydrationCtxs`	**Field Type** CtxAttrHydrationCtx[] **Description** Query that fetches data for a chosen attribute from the input schema for context-to-context mapping.
`inheritedFrom`	**Field Type** string **Description** Name of the parent context attribute mapping that's used to derive the current context attribute mapping.

ContextAttrHydrationDetail

Represents the SOQL (database) queries that fetch data for a chosen attribute from the input schema.

Field Name	Description
`contextAttrHydrationDetails`	**Field Type** ContextAttrHydrationDetail[]

Field Name	Description
	Description Details of the query that fetches the data for the specific query attribute.
`inheritedFrom`	**Field Type** string **Description** Name of the parent context attribute hydration detail that's used to derive the current context attribute hydration detail.
`objectName`	**Field Type** string **Description** Required. Name of the object used for the attribute hydration detail.
`queryAttribute`	**Field Type** string **Description** Required. The SOQL query that is the source of the hydration.

CtxAttrHydrationCtx

Represents the queries that fetch data for a chosen attribute from the input schema for context-to-context mapping.

Field Name	Description
`contextQueryAttribute`	**Field Type** string **Description** Required. Attribute in context definition that's the source of context hydration.
`inheritedFrom`	**Field Type** string **Description** Name of the parent context attribute hydration detail that's used to derive the current context attribute.

ContextNodeAttrDictionary

Represents the relationship between a context node and the context attribute dictionary.

Field Name	Description
`contextAttrDictIdentifier`	**Field Type** string **Description** Required. Developer name of the context attribute dictionary.
`contextNodeTagPrefix`	**Field Type** string **Description** Required. Tag prefix of the context node that's used to create the unique identifier of the parent context node.

ContextNode

Represents details of the structure of the nodes within the context. Each node can have other nodes related to them and attributes to describe the object. You can also define a hierarchy for the nodes.

Field Name	Description
`canonicalNode`	**Field Type** string **Description** Canonical node that's associated with the context node.
`contextAttributes`	**Field Type** ContextAttribute[] **Description** Details of the attribute used to describe the context node.
`contextNodeAttrDictionaries`	**Field Type** ContextNodeAttrDictionary[] **Description** Facilitates relationships between context node mapping and context dictionary. Additionally, it records the relationship between context node and context dictionary.
`contextTags`	**Field Type** ContextTag[]

Field Name	Description
	Description Unique identifier of the attribute or node.
`displayName`	**Field Type** string **Description** Display name of the context node.
`inheritedFrom`	**Field Type** string **Description** Name of the parent context node that's used to derive the current context node.
`title`	**Field Type** string **Description** Required. Name of the context node.
`transposable`	**Field Type** boolean **Description** Indicates whether the data in the Context Node record can be converted to field names (`true`) or not (`false`). The default value is `false`.

ContextAttribute

Represents details of an attribute used to describe a context node. Each node can have one or many associated attributes.

Field Name	Description
`contextTags`	**Field Type** ContextTag[] **Description** Shortened name of the attribute or node.
`dataType`	**Field Type** ContextAttributeDataType (enumeration of type string) **Description** Required.

Field Name	Description
	Type of data that's stored in the context attribute. Valid values are: • `boolean` • `currency` • `date` • `datetime` • `number` • `percent` • `picklist` • `reference` • `string` • `selfreference`—Available in API version 63.0 and later.
`description`	**Field Type** string **Description** Description of the context attribute.
`displayName`	**Field Type** string **Description** Display name of the context attribute.
`domainSet`	**Field Type** string **Description** List of node references to show the parent-child relationship between the nodes in a definition.
`fieldType`	**Field Type** ContextAttributeFieldType (enumeration of type string) **Description** Required. List of node references to depict the parent-child relation between the nodes in a definition. Valid values are: • `aggregate` • `input` • `inputoutput` • `output`

Field Name	Description
`inheritedFrom`	**Field Type** string **Description** Name of the parent attribute that's used to derive the current attribute.
`key`	**Field Type** boolean **Description** Indicates whether the attribute is a key attribute in the node (`true`) or not (`false`). The default value is `false`.
`title`	**Field Type** string **Description** Required. Name of the context attribute.
`transient`	**Field Type** boolean **Description** Indicates if an attribute is skipped in context persistence (`true`) or not (`false`). Available in API version 63.0 and later. The default value is `false`.
`value`	**Field Type** boolean **Description** Indicates whether the attribute identifies as a value in a node (`true`) or not (`false`). The default value is `false`.

ContextTag

Represents a unique identifier of an attribute or node instead of a fully qualified tag structure name.

Field Name	Description
`title`	**Field Type** string **Description** Required.

Field Name	Description
	Name of the context tag.
inheritedFrom	**Field Type** string **Description** Name of the parent context tag that's used to derive the current context tag.

Declarative Metadata Sample Definition

The following is an example of a ContextDefinition component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ContextDefinition xmlns="http://soap.sforce.com/2006/04/metadata">
    <fullName>Test</fullName>
    <contextDefinitionVersions>
        <contextMappings>
            <contextNodeMappings>
                <contextNodeAttrDictionaries>
                    <contextAttrDictIdentifier>Context Attribute Dictionary
Name</contextAttrDictIdentifier>
                    <contextNodeTagPrefix>Context Node Tag Prefix</contextNodeTagPrefix>
                </contextNodeAttrDictionaries>
                <contextAttributeMappings>
                    <contextAttrHydrationDetails>
                        <objectName>CustomAccount__c</objectName>
                        <queryAttribute>Name</queryAttribute>

<inheritedFrom>StandardDefinition/version/CustomAccountMapping/Praneeth/AccountName/hydrationInfo-1</inheritedFrom>

                    </contextAttrHydrationDetails>
                    <ctxAttrHydrationCtxs>
                        <contextQueryAttribute>StandardDefinition</contextQueryAttribute>


<inheritedFrom>StandardDefinition/version/AccountMapping/Praneeth/AccountName/ctxToCtxhydrationInfo-1</inheritedFrom>

                    </ctxAttrHydrationCtxs>
                    <contextAttribute>AccountName</contextAttribute>
                    <contextInputAttributeName>AccountName</contextInputAttributeName>

<inheritedFrom>StandardDefinition/version/CustomAccountMapping/Praneeth/AccountName</inheritedFrom>

                </contextAttributeMappings>
                <contextAttributeMappings>
                    <contextAttrHydrationDetails>
                        <objectName>CustomAccount__c</objectName>
                        <queryAttribute>CustomAccountName__c</queryAttribute>

<inheritedFrom>StandardDefinition/version/CustomAccountMapping/Praneeth/CustomAccountName/hydrationInfo-1</inheritedFrom>
```

```
                    </contextAttrHydrationDetails>
                    <ctxAttrHydrationCtxs>
                        <contextQueryAttribute>StandardDefinition</contextQueryAttribute>

<inheritedFrom>StandardDefinition/version/AccountMapping/Praneeth/AccountName/ctxToCtxhydrationInfo-1</inheritedFrom>

                    </ctxAttrHydrationCtxs>
                    <contextAttribute>CustomAccountName</contextAttribute>
                <contextInputAttributeName>CustomAccountName</contextInputAttributeName>

<inheritedFrom>StandardDefinition/version/CustomAccountMapping/Praneeth/CustomAccountName</inheritedFrom>

                </contextAttributeMappings>
                <contextNode>Praneeth</contextNode>
                <object>CustomAccount__c</object>

<inheritedFrom>StandardDefinition/version/CustomAccountMapping/Praneeth</inheritedFrom>
              <mappedContextDefinition>CustomContextDefinition</mappedContextDefinition>

            </contextNodeMappings>
            <contextMappingIntents>
                <mappingIntent>hydration</mappingIntent>
            </contextMappingIntents>
            <default>true</default>
            <title>CustomAccountMapping</title>
           <inheritedFrom>StandardDefinition/version/CustomAccountMapping</inheritedFrom>

        </contextMappings>
        <contextMappings>
            <contextNodeMappings>
                <contextNodeAttrDictionaries>
                    <contextAttrDictIdentifier>Context Attribute Dictionary
Name</contextAttrDictIdentifier>
                    <contextNodeTagPrefix>Context Node Tag Prefix</contextNodeTagPrefix>
                </contextNodeAttrDictionaries>
                <contextAttributeMappings>
                    <contextAttrHydrationDetails>
                        <objectName>Account</objectName>
                        <queryAttribute>Name</queryAttribute>

<inheritedFrom>StandardDefinition/version/AccountMapping/Praneeth/CustomAccountName/AccountName/hydrationInfo-1</inheritedFrom>

                    </contextAttrHydrationDetails>
                    <ctxAttrHydrationCtxs>
                        <contextQueryAttribute>StandardDefinition</contextQueryAttribute>

<inheritedFrom>StandardDefinition/version/AccountMapping/Praneeth/AccountName/ctxToCtxhydrationInfo-1</inheritedFrom>

                    </ctxAttrHydrationCtxs>
                    <contextAttribute>AccountName</contextAttribute>
                    <contextInputAttributeName>AccountName</contextInputAttributeName>
```

```
<inheritedFrom>StandardDefinition/version/AccountMapping/Praneeth/CustomAccountName/AccountName</inheritedFrom>

                </contextAttributeMappings>
                <contextAttributeMappings>
                    <contextAttrHydrationDetails>
                        <objectName>Account</objectName>
                        <queryAttribute>CustomAccountName__c</queryAttribute>

<inheritedFrom>StandardDefinition/version/AccountMapping/Praneeth/CustomAccountName/hydrationInfo-1</inheritedFrom>

                    </contextAttrHydrationDetails>
                    <ctxAttrHydrationCtxs>
                        <contextQueryAttribute>StandardDefinition</contextQueryAttribute>


<inheritedFrom>StandardDefinition/version/AccountMapping/Praneeth/AccountName/ctxToCtxhydrationInfo-1</inheritedFrom>

                    </ctxAttrHydrationCtxs>
                    <contextAttribute>CustomAccountName</contextAttribute>
                 <contextInputAttributeName>CustomAccountName</contextInputAttributeName>


<inheritedFrom>StandardDefinition/version/AccountMapping/Praneeth/CustomAccountName</inheritedFrom>

                </contextAttributeMappings>
                <contextNode>Praneeth</contextNode>
                <object>Account</object>

<inheritedFrom>StandardDefinition/version/AccountMapping/Praneeth</inheritedFrom>
               <mappedContextDefinition>CustomContextDefinition</mappedContextDefinition>

            </contextNodeMappings>
             <contextMappingIntents>
                <mappingIntent>persistence</mappingIntent>
             </contextMappingIntents>
            <description>Account Mapping</description>
            <default>false</default>
            <title>AccountMapping</title>
            <inheritedFrom>StandardDefinition/version/AccountMapping</inheritedFrom>
        </contextMappings>
        <contextNodes>
            <contextNodeAttrDictionaries>
                <contextAttrDictIdentifier>Context Attribute Dictionary
Name</contextAttrDictIdentifier>
                <contextNodeTagPrefix>Context Node Tag Prefix</contextNodeTagPrefix>
            </contextNodeAttrDictionaries>
            <contextAttributes>
                <contextTags>
                    <title>AccountName</title>

<inheritedFrom>StandardDefinition/version/Praneeth/AccountName/AccountName</inheritedFrom>

                </contextTags>
```

```
                <dataType>string</dataType>
                <fieldType>inputoutput</fieldType>
                <key>false</key>
                <title>AccountName</title>
                <displayName>AccountName</displayName>
                <description>Test Description</description>
                <value>false</value>

<inheritedFrom>StandardDefinition/version/Praneeth/AccountName</inheritedFrom>
            </contextAttributes>
            <contextAttributes>
                <dataType>string</dataType>
                <fieldType>inputoutput</fieldType>
                <key>false</key>
                <title>CustomAccountName</title>
                <value>false</value>
                <displayName>CustomAccountName</displayName>
                <description>Test Description</description>

<inheritedFrom>StandardDefinition/version/Praneeth/CustomAccountName</inheritedFrom>
            </contextAttributes>
            <contextTags>
                <title>Praneeth</title>
              <inheritedFrom>StandardDefinition/version/Praneeth/Praneeth</inheritedFrom>

            </contextTags>
            <title>Praneeth</title>
            <transposable>false</transposable>
            <inheritedFrom>StandardDefinition/version/Praneeth</inheritedFrom>
            <canonicalNode></canonicalNode>
            <displayName>Praneeth</displayName>
        </contextNodes>
        <endDate>2097-05-10 00:00:00</endDate>
        <startDate>2023-05-10 00:00:00</startDate>
        <versionNumber>1</versionNumber>
        <isActive>true</isActive>
    </contextDefinitionVersions>
    <description>Test Description</description>
    <contextTtl>10</contextTtl>
    <inheritedFrom>StandardDefinition</inheritedFrom>
    <inheritedFromVersion>1.0</inheritedFromVersion>
    <clonedFrom>OriginalDefinition</clonedFrom>
    <isProtected>false</isProtected>
    <masterLabel>Test Label</masterLabel>
    <title>TestTitle</title>
    <displayName>TestTitle</displayName>
</ContextDefinition>
```

The following is an example `package.xml` that references the previous definition.

```
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>Test</members>
        <name>ContextDefinition</name>
    </types>
```

```
    <types>
        <members>Account.CustomAccountName__c</members>
        <name>CustomField</name>
    </types>
    <types>
        <members>CustomAccount__c</members>
        <name>CustomObject</name>
    </types>
    <version>60.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ConversationMessageDefinition

Represents a messaging component in an Enhanced Messaging channel or Messaging for In-App and Web session.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

`ConversationMessageDefinition` components have the suffix `.conversationMessageDefinition` and are stored in the `conversationMessageDefinitions` folder.

Version

`ConversationMessageDefinition` is supported for use in enhanced Messaging channels and Messaging for In-App and Web, and is available in API version 59.0 and later.

Fields

Field Name	Description
`constants`	**Field Type** ConversationMessageConstant[] **Description** An array of constants that defines the messaging components. Constants support multiple data types, including text, URL, and image.
`description`	**Field Type** string

Field Name	Description
	Description The description of the conversation message definition.
`label`	**Field Type** string **Description** A user-friendly name for `ConversationMessageDefinition`, which is defined when `ConversationMessageDefinition` is created.
`language`	**Field Type** string **Description** The language of the conversation message definition.
`messageHandlers`	**Field Type** ConversationMessageHandler[] **Description** An array of message handlers.
`messageLayouts`	**Field Type** ConversationMessageLayout[] **Description** An array of message layouts.
`optionsParameter`	**Field Type** ConversationMessageOptionsParameter[] **Description** An array of options parameter of the `ConversationMessageDefinition`.
`parameters`	**Field Type** ConversationMessageParameter[] **Description** An array of parameters.
`type`	**Field Type** ConversationMessageDefinitionType (enumeration of type string) **Description** Required. The type of the conversation message definition. Valid values are: • `Action` • `AuthenticationRequest` • `AutoResponse`

Field Name	Description
	• `Link` • `Notification` • `PaymentRequest` • `Picklist` • `RecordPicker` • `RecordView` • `TimePicker`

ConversationMessageConstant

Represents a constant value on the messaging component. When a messaging component is created in the UI, the text and images entered during creation are saved as standard constants. Custom constants can also be added.

Field Name	Description
`compositeValues`	**Field Type** ConversationMessageConstantCompositeValue[] **Description** An array of composite values of `ConversationMessageConstant`.
`constantType`	**Field Type** ConversationMessageConstantType (enumeration of type string) **Description** Required. The conversation message constant type. Valid values are: • `Custom` • `Image` • `Options` • `SubTitle` • `Title` • `Url`
`label`	**Field Type** string **Description** The UI label of the conversation message constant.
`name`	**Field Type** string **Description** The name of the conversation message constant.

Field Name	Description
`primitiveValues`	**Field Type** ConversationMessageConstantPrimitiveValue (enumeration of type string) **Description** An array of primitive values of `ConversationMessageConstant`.
`valueType`	**Field Type** ConversationMessageValueType (enumeration of type string) **Description** The type of the conversation message constant value. Valid values are: • `Boolean` • `Date` • `DateTime` • `Double` • `ImageId` • `Integer` • `RecordId` • `Text` • `Url`

ConversationMessageConstantCompositeValue

Represents the composite values of the ConversationMessageConstant.

Field Name	Description
`constantItems`	**Field Type** ConversationMessageConstant[] **Description** An array of constant items.
`identifier`	**Field Type** string **Description** Required. The client identifier.

ConversationMessageConstantPrimitiveValue

Represents the primitive values of the ConversationMessageConstant.

Field Name	Description
`contentAssetName`	**Field Type** string **Description** Represents the value for type = ImageAsset
`textValue`	**Field Type** string **Description** Represents the value for type = Text
`type`	**Field Type** ConversationMessageConstantValueType (enumeration of type string) **Description** Required. The type of the conversation message constant primitive value. Valid values are: • `ImageAsset` • `Text` • `Url`
`urlValue`	**Field Type** string **Description** Represents the value for type = Url

ConversationMessageHandler

Represents the conversation message handler.

Field Name	Description
`activeRequestDurationMinutes`	**Field Type** int **Description** Required. The duration of an active request in minutes.
`handlerName`	**Field Type** string **Description** Required. The name of the message handler.

Field Name	Description
handlerType	**Field Type** ConversationMessageHandlerType (enumeration of type string) **Description** Required. The type of message handler. Valid values are: • AuthProvider • PaymentProvider • QuickAction

ConversationMessageLayout

Represents the conversation message layout.

Field Name	Description
externalTemplates	**Field Type** ConvMsgExternalTemplateVersion[] **Description** The external template version of the ConversationMessageLayout.
formatType	**Field Type** ConversationMessageFormatType (enumeration of type string) **Description** Required. The format type of the conversation message layout. Valid values are: • Application • Buttons • Carousel • EncryptedOAuthToken • ExternalTemplate • Inputs • ListPicker • Media • Payment • QuickReplies • RichLink • Text • TimePicker • WebView

Field Name	Description
`layoutItems`	**Field Type** ConversationMessageLayoutItem[] **Description** An array of layout items.
`messageType`	**Field Type** ConversationMessageType (enumeration of type string) **Description** Required. The conversation message type. Valid values are: • `AuthenticationRequest` • `Choices` • `Form` • `PaymentRequest` • `StaticContent`

ConvMsgExternalTemplateVersion

Represents the external template version of the conversation message layout.

Field Name	Description
`accountIdentifier`	**Field Type** string **Description** Required. The account identifier. For WhatsApp channels, this is the WABA ID.
`accountName`	**Field Type** string **Description** Required. The account name.
`language`	**Field Type** string **Description** Required. The language of the conversation message external template.
`status`	**Field Type** ConvMsgExternalTemplateVersionStatus (enumeration of type string) **Description** Required. The status of the conversation message external template. Valid values are:

Field Name	Description
	• `Approved` • `Disabled` • `InAppeal` • `Paused` • `Pending` • `Rejected`
`templateName`	**Field Type** string **Description** Required. The name of the conversation message external template.
`templateVersionIdentifier`	**Field Type** string **Description** Required. The template version identifier.

ConversationMessageLayoutItem

Represents the conversation message layout item.

Field Name	Description
`collectionType`	**Field Type** ConversationMessageCollectionType (enumeration of type string) **Description** Required. The type of conversation message collection. Valid values are: • `DynamicList` • `None` • `StaticList`
`compositeValues`	**Field Type** ConversationMessageLayoutCompositeValue[] **Description** An array of composite values of the `ConversationMessageLayoutItem`.
`name`	**Field Type** string **Description** The name of the conversation message layout item.

Field Name	Description
`primitiveValues`	**Field Type** ConversationMessageLayoutPrimitiveValue[] **Description** An array of primitive values of the `ConversationMessageLayoutItem`.

ConversationMessageLayoutCompositeValue

Represents the composite value of the ConversationMessageLayoutItem.

Field Name	Description
`compositeTypeName`	**Field Type** string **Description** Required. The name of the conversation message layout composite value type.
`layoutItems`	**Field Type** ConversationMessageLayoutItem[] **Description** An array of layout items.
`valueSourceReference`	**Field Type** string **Description** The source of the conversation message layout composite value.

ConversationMessageLayoutPrimitiveValue

Represents the primitive value of the ConversationMessageLayoutItem.

Field Name	Description
`contentAssetName`	**Field Type** string **Description** The content asset name.
`fieldName`	**Field Type** string **Description** The name of the conversation message layout primitive value field.

Field Name	Description
`formulaTemplate`	**Field Type** string **Description** The formula template defines the content for each entry in the list.
`literalValue`	**Field Type** string **Description** The literal primitive value of the conversation message layout.
`mergeFields`	**Field Type** ConversationMessageMergeField[] **Description** Inserts multiple values to a list.
`type`	**Field Type** ConversationMessageLayoutValueType (enumeration of type string) **Description** Required. The type of the conversation message layout primitive value. Valid values are: • `FormulaTemplate` • `Literal` • `MediaAsset` • `SourcePrimitiveValue` • `SourceSobjectField` • `SourceSobjectFieldValue` • `SourceSobjectFormula`
`valueFormula`	**Field Type** string **Description** The formula of the conversation message layout primitive value.
`valueSourceReference`	**Field Type** string **Description** The source of the conversation message layout primitive value.

ConversationMessageMergeField

Merge field is used to insert multiple values to a list.

Field Name	Description
`formulaTemplate`	**Field Type** string **Description** Required. The formula template of the conversation message merge field.
`mergeFieldType`	**Field Type** ConversationMessageMergeFieldType (enumeration of type string) **Description** Required. The type of the conversation message merge field. Valid value is ListTemplate.
`name`	**Field Type** string **Description** Required. The name of the conversation message merge field.
`valueSourceReference`	**Field Type** string **Description** Required. The source of the conversation message merge field value.

ConversationMessageOptionsParameter

Represents a conversation message options parameter.

Field Name	Description
`compositeTypeDetails`	**Field Type** ConversationMessageParameterCompositeDetails[] **Description** An array of composite details of `ConversationMessageOptionsParameter`.
`optionsParameterType`	**Field Type** ConversationMessageOptionsParameterType (enumeration of type string) **Description** Required. The type of conversation message options parameter. Valid values are: • `CustomCompositeOptions` • `CustomPrimitiveOptions` • `RecordIdOptions`

Field Name	Description
	• `TimeSlotOptions`
`primitiveTypeDetails`	**Field Type** ConversationMessageParameterPrimitiveDetails **Description** The primitive type details of conversation message options parameter.

ConversationMessageParameterCompositeDetails

Represents the composite details of a conversation message parameter.

Field Name	Description
`compositeChildItems`	**Field Type** ConversationMessageParameterCompositeDetails[] **Description** The composite child items of the conversation message parameter.
`isList`	**Field Type** boolean **Description** Indicates whether the conversation message parameter composite details field is a list item (`true`) or not (`false`). The default value is false.
`isRequired`	**Field Type** boolean **Description** Indicates whether the conversation message parameter is required (`true`) or not (`false`). The default value is false.
`label`	**Field Type** string **Description** The UI label of the conversation message parameter composite details field.
`maxListItems`	**Field Type** int **Description** The maximum number of list items in the conversation message parameter composite details field.

Field Name	Description
`name`	**Field Type** string **Description** The name of the conversation message parameter composite details field.
`primitiveChildItems`	**Field Type** ConversationMessageParameterPrimitiveDetails[] **Description** An array of primitive child items.

ConversationMessageParameterPrimitiveDetails

Represents the primitive details of the conversation message parameter.

Field Name	Description
`isList`	**Field Type** boolean **Description** Indicates whether the conversation message parameter primitive details field is a list item (`true`) or not (`false`). The default value is false.
`isRequired`	**Field Type** boolean **Description** Indicates whether the conversation message parameter primitive details field is required (`true`) or not (`false`). The default value is false.
`label`	**Field Type** string **Description** The UI label of the conversation message parameter primitive details field.
`maxListItems`	**Field Type** int **Description** The maximum number of list items that are allowed in the conversation message parameter primitive details field.
`name`	**Field Type** string

Field Name	Description
	Description The name of the conversation message parameter primitive details field.
`sobjectType`	**Field Type** string **Description** The sObject type.
`valueType`	**Field Type** ConversationMessageValueType (enumeration of type string) **Description** The type of the conversation message parameter value. Valid values are: • `Boolean` • `Date` • `DateTime` • `Double` • `ImageId` • `Integer` • `RecordId` • `Text` • `Url`

ConversationMessageParameter

Represents a conversation message parameter.

Field Name	Description
`compositeTypeDetails`	**Field Type** ConversationMessageParameterCompositeDetails **Description** An array of composite type details.
`parameterType`	**Field Type** ConversationMessageParameterType (enumeration of type string) **Description** Required. The type of conversation message parameter. Valid values are: • `CustomComposite` • `CustomPrimitive` • `RecordIds`

Field Name	Description
primitiveTypeDetails	**Field Type** ConversationMessageParameterPrimitiveDetails **Description** An array of primitive type details.

Declarative Metadata Sample Definition

The following is an example of a `ConversationMessageDefinition` component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ConversationMessageDefinition xmlns="http://soap.sforce.com/2006/04/metadata">
    <constants>
        <constantType>Custom</constantType>
        <label>imageAsset</label>
        <name>imageAsset</name>
        <primitiveValues>
            <contentAssetName>Screenshot_20240402_at_32437PM</contentAssetName>
            <type>ImageAsset</type>
        </primitiveValues>
        <valueType>ImageId</valueType>
    </constants>
    <constants>
        <constantType>Custom</constantType>
        <label>message</label>
        <name>message</name>
        <primitiveValues>
            <textValue>Favourite Season</textValue>
            <type>Text</type>
        </primitiveValues>
        <valueType>Text</valueType>
    </constants>
    <constants>
        <constantType>Custom</constantType>
        <label>Prompt1</label>
        <name>Prompt1</name>
        <primitiveValues>
            <textValue>Choose one option</textValue>
            <type>Text</type>
        </primitiveValues>
        <valueType>Text</valueType>
    </constants>
    <constants>
        <compositeValues>
            <constantItems>
                <constantType>Image</constantType>
                <primitiveValues>
                    <contentAssetName>Screenshot_20240321_at_53957PM3</contentAssetName>
                    <type>ImageAsset</type>
                </primitiveValues>
            </constantItems>
```

```
            <constantItems>
                <constantType>SubTitle</constantType>
                <primitiveValues>
                    <textValue>January</textValue>
                    <type>Text</type>
                </primitiveValues>
            </constantItems>
            <constantItems>
                <constantType>Title</constantType>
                <primitiveValues>
                    <textValue>Jan</textValue>
                    <type>Text</type>
                </primitiveValues>
            </constantItems>
            <identifier>1c6f8c4d-7bce-1649-fa45-db587bcfbb29</identifier>
        </compositeValues>
        <compositeValues>
            <constantItems>
                <constantType>Image</constantType>
                <primitiveValues>
                    <contentAssetName>Screenshot_20240321_at_53957PM4</contentAssetName>
                    <type>ImageAsset</type>
                </primitiveValues>
            </constantItems>
            <constantItems>
                <constantType>SubTitle</constantType>
                <primitiveValues>
                    <textValue>December</textValue>
                    <type>Text</type>
                </primitiveValues>
            </constantItems>
            <constantItems>
                <constantType>Title</constantType>
                <primitiveValues>
                    <textValue>Dec</textValue>
                    <type>Text</type>
                </primitiveValues>
            </constantItems>
            <identifier>fb8bb328-7bc7-2830-6194-2ae7ece055ad</identifier>
        </compositeValues>
        <compositeValues>
            <constantItems>
                <constantType>Image</constantType>
                <primitiveValues>
                    <contentAssetName>Screenshot_20240321_at_53912PM1</contentAssetName>
                    <type>ImageAsset</type>
                </primitiveValues>
            </constantItems>
            <constantItems>
                <constantType>SubTitle</constantType>
                <primitiveValues>
                    <textValue>March</textValue>
                    <type>Text</type>
                </primitiveValues>
```

```
            </constantItems>
            <constantItems>
                <constantType>Title</constantType>
                <primitiveValues>
                    <textValue>March</textValue>
                    <type>Text</type>
                </primitiveValues>
            </constantItems>
            <identifier>570baa88-fa4d-4b31-0e84-92f87b35af0a</identifier>
        </compositeValues>
        <constantType>Options</constantType>
    </constants>
    <constants>
        <constantType>Title</constantType>
        <primitiveValues>
            <textValue>What is your favourite month?</textValue>
            <type>Text</type>
        </primitiveValues>
    </constants>
    <label>Favourite Month</label>
    <language>en_US</language>
    <messageLayouts>
        <formatType>Buttons</formatType>
        <layoutItems>
            <collectionType>DynamicList</collectionType>
            <compositeValues>
                <compositeTypeName>TitleOptionItem</compositeTypeName>
                <layoutItems>
                    <collectionType>None</collectionType>
                    <compositeValues>
                        <compositeTypeName>TitleItem</compositeTypeName>
                        <layoutItems>
                            <collectionType>None</collectionType>
                            <name>title</name>
                            <primitiveValues>
                                <type>SourcePrimitiveValue</type>

<valueSourceReference>Constants.Options.ListItem.SubTitle</valueSourceReference>
                            </primitiveValues>
                        </layoutItems>
                    </compositeValues>
                    <name>titleItem</name>
                </layoutItems>
                <valueSourceReference>Constants.Options</valueSourceReference>
            </compositeValues>
            <name>optionItems</name>
        </layoutItems>
        <layoutItems>
            <collectionType>None</collectionType>
            <name>text</name>
            <primitiveValues>
                <type>SourcePrimitiveValue</type>
                <valueSourceReference>Constants.Title</valueSourceReference>
            </primitiveValues>
```

```
        </layoutItems>
        <messageType>Choices</messageType>
    </messageLayouts>
    <messageLayouts>
        <formatType>ListPicker</formatType>
        <layoutItems>
            <collectionType>None</collectionType>
            <compositeValues>
                <compositeTypeName>TitleImageItem</compositeTypeName>
                <layoutItems>
                    <collectionType>None</collectionType>
                    <name>imageId</name>
                    <primitiveValues>
                        <type>SourcePrimitiveValue</type>
                        <valueSourceReference>Constants.imageAsset</valueSourceReference>

                    </primitiveValues>
                </layoutItems>
                <layoutItems>
                    <collectionType>None</collectionType>
                    <name>title</name>
                    <primitiveValues>
                        <type>SourcePrimitiveValue</type>
                        <valueSourceReference>Constants.Title</valueSourceReference>
                    </primitiveValues>
                </layoutItems>
            </compositeValues>
            <name>message</name>
        </layoutItems>
        <layoutItems>
            <collectionType>DynamicList</collectionType>
            <compositeValues>
                <compositeTypeName>TitleOptionItem</compositeTypeName>
                <layoutItems>
                    <collectionType>None</collectionType>
                    <compositeValues>
                        <compositeTypeName>TitleImageItem</compositeTypeName>
                        <layoutItems>
                            <collectionType>None</collectionType>
                            <name>imageId</name>
                            <primitiveValues>
                                <type>SourcePrimitiveValue</type>

<valueSourceReference>Constants.Options.ListItem.Image</valueSourceReference>
                            </primitiveValues>
                        </layoutItems>
                        <layoutItems>
                            <collectionType>None</collectionType>
                            <name>title</name>
                            <primitiveValues>
                                <type>SourcePrimitiveValue</type>

<valueSourceReference>Constants.Options.ListItem.Title</valueSourceReference>
                            </primitiveValues>
```

```
                    </layoutItems>
                </compositeValues>
                <name>titleItem</name>
            </layoutItems>
            <valueSourceReference>Constants.Options</valueSourceReference>
        </compositeValues>
        <name>optionItems</name>
    </layoutItems>
    <layoutItems>
        <collectionType>None</collectionType>
        <compositeValues>
            <compositeTypeName>TitleImageItem</compositeTypeName>
            <layoutItems>
                <collectionType>None</collectionType>
                <name>imageId</name>
                <primitiveValues>
                    <type>SourcePrimitiveValue</type>
                    <valueSourceReference>Constants.imageAsset</valueSourceReference>

                </primitiveValues>
            </layoutItems>
            <layoutItems>
                <collectionType>None</collectionType>
                <name>title</name>
                <primitiveValues>
                    <type>SourcePrimitiveValue</type>
                    <valueSourceReference>Constants.message</valueSourceReference>
                </primitiveValues>
            </layoutItems>
        </compositeValues>
        <name>reply</name>
    </layoutItems>
    <layoutItems>
        <collectionType>None</collectionType>
        <name>title</name>
        <primitiveValues>
            <type>SourcePrimitiveValue</type>
            <valueSourceReference>Constants.Title</valueSourceReference>
        </primitiveValues>
    </layoutItems>
    <messageType>Choices</messageType>
</messageLayouts>
<messageLayouts>
    <formatType>Carousel</formatType>
    <layoutItems>
        <collectionType>DynamicList</collectionType>
        <compositeValues>
            <compositeTypeName>TitleItemWithInteractions</compositeTypeName>
            <layoutItems>
                <collectionType>StaticList</collectionType>
                <compositeValues>
                    <compositeTypeName>TitleOptionItem</compositeTypeName>
                    <layoutItems>
                        <collectionType>None</collectionType>
```

```
                            <compositeValues>
                                <compositeTypeName>TitleItem</compositeTypeName>
                                <layoutItems>
                                    <collectionType>None</collectionType>
                                    <name>title</name>
                                    <primitiveValues>
                                        <literalValue>Select One</literalValue>
                                        <type>Literal</type>
                                    </primitiveValues>
                                </layoutItems>
                            </compositeValues>
                            <name>titleItem</name>
                        </layoutItems>
                    </compositeValues>
                    <name>interactionItems</name>
                </layoutItems>
                <layoutItems>
                    <collectionType>None</collectionType>
                    <compositeValues>
                        <compositeTypeName>TitleImageItem</compositeTypeName>
                        <layoutItems>
                            <collectionType>None</collectionType>
                            <name>imageId</name>
                            <primitiveValues>
                                <type>SourcePrimitiveValue</type>

<valueSourceReference>Constants.Options.ListItem.Image</valueSourceReference>
                            </primitiveValues>
                        </layoutItems>
                        <layoutItems>
                            <collectionType>None</collectionType>
                            <name>subTitle</name>
                            <primitiveValues>
                                <type>SourcePrimitiveValue</type>

<valueSourceReference>Constants.Options.ListItem.SubTitle</valueSourceReference>
                            </primitiveValues>
                        </layoutItems>
                        <layoutItems>
                            <collectionType>None</collectionType>
                            <name>title</name>
                            <primitiveValues>
                                <type>SourcePrimitiveValue</type>
                                 <valueSourceReference>Constants.Title</valueSourceReference>

                            </primitiveValues>
                        </layoutItems>
                    </compositeValues>
                    <name>titleItem</name>
                </layoutItems>
                <valueSourceReference>Constants.Options</valueSourceReference>
            </compositeValues>
            <name>items</name>
        </layoutItems>
```

```
        <messageType>Choices</messageType>
    </messageLayouts>
    <messageLayouts>
        <formatType>Text</formatType>
        <layoutItems>
            <collectionType>None</collectionType>
            <name>text</name>
            <primitiveValues>
                <formulaTemplate>{!$Constants.Title}
{!$Constants.Prompt1}:
{!$ListTemplates.OptionsList}</formulaTemplate>
                <mergeFields>
                    <formulaTemplate>{!$ListItem.Index}.
{!$ListItem.Value.Title}{!BR()}</formulaTemplate>
                    <mergeFieldType>ListTemplate</mergeFieldType>
                    <name>OptionsList</name>
                    <valueSourceReference>Constants.Options</valueSourceReference>
                </mergeFields>
                <type>FormulaTemplate</type>
            </primitiveValues>
        </layoutItems>
        <messageType>StaticContent</messageType>
    </messageLayouts>
    <type>Picklist</type>
</ConversationMessageDefinition>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>Favourite_Month</members>
        <name>ConversationMessageDefinition</name>
    </types>
    <version>61.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ConversationVendorInfo

Represents the connection between the partner vendor system and the Service Cloud feature. For example, for Service Cloud Voice, this type contains information about the partner telephony system or Contact Center as a Service (CCaaS) system. For Bring Your Own Channel for Messaging or Bring Your Own Channel for CCaaS, this type contains information about the partner messaging system or CCaaS system.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

ConversationVendorInfo components have the suffix `.ConversationVendorInformation` and are stored in the `ConversationVendorInformation` folder.

Version

ConversationVendorInfo components are available in API version 52.0 and later.

Special Access Rules

This type requires an add-on license for Service Cloud Voice for Partner Telephony or Digital Engagement.

Fields

The fields in the ConversationVendorInfo type apply to all Service Cloud features unless otherwise stated in the field description. For example, if a field applies to just one Service Cloud Voice telephony model setup or is applied differently by different partner systems, this is stated in the field description.

Field Name	Description
`agentSSOSupported`	**Field Type** boolean **Description** If set to `true`, agents can single sign-on (SSO) into their contact center using Salesforce as the identity provider (IdP). Behind the scenes, Salesforce is used as the SAML IdP in the Single Sign-On connected app for the contact center. If set to `false`, an IdP other than Salesforce is used or an IdP isn't used at all. The default value is `false`. If this value is set to `false` and you want to use Salesforce as the IdP for your contact center, set this value and the `namedCredentialSupported` value to `true` and configure the `service_cloud_voice.PartnerSSO` interface in your Apex integration class. Available in API version 53.0 and later. Applies to the following implementations: • Service Cloud Voice with Partner Telephony • Service Cloud Voice with Partner Telephony from Amazon Connect • Bring Your Own Channel for CCaaS
`awsAccountKey`	**Field Type** string

Field Name	Description
	Description The 12-digit AWS subaccount ID that's automatically provisioned for you when Service Cloud Voice was turned on. Available in API version 55.0 and later. Applies to the following implementation: • Service Cloud Voice with Amazon Connect
`awsRootEmail`	**Field Type** string **Description** The email address used by Salesforce to create the root user for the provisioned AWS subaccount when Service Cloud Voice was turned on. Available in API version 55.0 and later. Applies to the following implementation: • Service Cloud Voice with Amazon Connect
`awsTenantVersion`	**Field Type** double **Description** The version number of the SVCTenantStack AWS CloudFormation stack that's deployed. The stack is deployed in AWS region "us-east-1". Available in API version 55.0 and later. Applies to the following implementation: • Service Cloud Voice with Amazon Connect
`bridgeComponent`	**Field Type** string **Description** The Lightning component used to communicate between the telephony or messaging system and other Lightning components. Applies to the following implementations: • Service Cloud Voice with Partner Telephony • Service Cloud Voice with Partner Telephony from Amazon Connect • Bring Your Own Channel for CCaaS
`clientAuthMode`	**Field Type** ClientAuthMode (enumeration of type string) **Description** The client authentication mode. Values are: • `Custom` • `Mixed`

Field Name	Description
	• SSO Applies to the following implementations: • Service Cloud Voice with Partner Telephony • Service Cloud Voice with Partner Telephony from Amazon Connect
connectorUrl	**Field Type** string **Description** The URL that hosts your Service Cloud Voice or Bring Your Own Channel for CCaaS connector. This value could be a Visualforce page or a public URL. Applies to the following implementations: • Service Cloud Voice with Partner Telephony • Service Cloud Voice with Partner Telephony from Amazon Connect • Bring Your Own Channel for CCaaS
customConfig	**Field Type** string **Description** The foreign key to the CustomEntityDefinition, which contains partner-specific custom settings. Available in API version 53.0 and later. Applies to the following implementations: • Service Cloud Voice with Partner Telephony • Service Cloud Voice with Partner Telephony from Amazon Connect • Bring Your Own Channel for CCaaS
customIcon	**Field Type** string **Description** ID of the static resource used to identify the contact center integration, such as a Contact Center as a Service (CCaaS) provider logo. The static resource must be in SVG format. This field is optional. Available in API version 62.0 and later. Applies to the following implementations: • Service Cloud Voice with Partner Telephony • Bring Your Own Channel for CCaaS
customLoginUrl	**Field Type** string **Description** The URL that hosts your telephony system or CCaaS system login page. Applies to the following implementations:

Field Name	Description
	• Service Cloud Voice with Partner Telephony • Service Cloud Voice with Partner Telephony from Amazon Connect • Bring Your Own Channel for CCaaS
`developerName`	**Field Type** string **Description** The unique name of the type in the API.
`einsteinConversationInsightsSupported`	**Field Type** boolean **Description** If set to `true`, Einstein Conversation Insights is turned on. The default value is `false`. Available in API version 53.0 and later. Applies to the following implementations: • Service Cloud Voice with Partner Telephony • Service Cloud Voice with Partner Telephony from Amazon Connect
`integrationClass`	**Field Type** string **Description** The foreign key to the partner Apex class implementing supported interfaces. Available in API version 53.0 and later. Applies to the following implementations: • Service Cloud Voice with Partner Telephony • Service Cloud Voice with Partner Telephony from Amazon Connect • Bring Your Own Channel for CCaaS
`integrationClassName`	**Field Type** string **Description** Deprecated in API version 53.0. Don't set this field. Instead, use `integrationClass`. Applies to the following implementations: • Service Cloud Voice with Partner Telephony • Service Cloud Voice with Partner Telephony from Amazon Connect
`intelligenceSupported`	**Field Type** boolean

Field Name	Description
	Description If set to `true`, Salesforce ingests real-time signals sent from a partner telephony system. If set to `false`, Salesforce won't ingest real-time intelligence signals from a partner telephony system. The default value is `false`. Available in API version 59.0 and later. Applies to the following implementations: • Service Cloud Voice with Partner Telephony
`isTaxCompliant`	**Field Type** boolean **Description** Indicates whether the Amazon tax settings for the AWS subaccount provisioned for Service Cloud Voice have been confirmed (`true`). The default value is `false`. Available in API version 55.0 and later. Applies to the following implementation: • Service Cloud Voice with Amazon Connect
`keyProvisioningSupported`	**Field Type** boolean **Description** If set to `true`, key provisioning and renewal are automated. The default value is `false`. Available in API version 54.0 and later. Applies to the following implementations: • Service Cloud Voice with Partner Telephony • Service Cloud Voice with Partner Telephony from Amazon Connect
`masterLabel`	**Field Type** string **Description** The partner vendor's display name as it appears in the UI. This name appears in several places in the UI, so include the partner vendor name for easy identification. For Service Cloud Voice, this label also represents the telephony provider name in the contact center record. For Service Cloud Voice with Amazon Connect, this field is always set to `Service Cloud Voice`.
`namedCredential`	**Field Type** string **Description** A sample-named credential that can be used for Apex callouts to the partner system. Available in API version 53.0 and later.

Field Name	Description
	Applies to the following implementations: • Service Cloud Voice with Partner Telephony • Service Cloud Voice with Partner Telephony from Amazon Connect • Bring Your Own Channel for CCaaS
`namedCredentialSupported`	**Field Type** boolean **Description** A sample-named credential that can be used for Apex callouts to the partner system. Available in API version 53.0 and later. Applies to the following implementations: • Service Cloud Voice with Partner Telephony • Service Cloud Voice with Partner Telephony from Amazon Connect • Bring Your Own Channel for CCaaS
`partnerContactCenterListSupported`	**Field Type** boolean **Description** If set to `true`, enables the customer to select one contact center from a list of multiple contact centers to connect with Salesforce. The default value is `false`. Available in API version 53.0 and later. Applies to the following implementations: • Service Cloud Voice with Partner Telephony • Service Cloud Voice with Partner Telephony from Amazon Connect • Bring Your Own Channel for CCaaS
`partnerPhoneNumbersSupported`	**Field Type** boolean **Description** If set to `true`, displays a list of phone numbers used to create contact center channels. The default value is `false`. Available in API version 54.0 and later. Applies to the following implementations: • Service Cloud Voice with Partner Telephony • Service Cloud Voice with Partner Telephony from Amazon Connect
`partnerTransferDestinationsSupported`	**Field Type** boolean **Description** If set to `true`, allows Salesforce to fetch contact center queues so that Salesforce and contact center queues can be mapped. The default value is `false`. Available in API version 53.0 and later.

Field Name	Description
	Applies to the following implementations: • Service Cloud Voice with Partner Telephony • Service Cloud Voice with Partner Telephony from Amazon Connect • Bring Your Own Channel for CCaaS
`queueManagementSupported`	**Field Type** boolean **Description** If set to `true`, support queue management. The default value is `false`. Available in API version 56.0 and later. Applies to the following implementations: • Service Cloud Voice with Partner Telephony • Service Cloud Voice with Partner Telephony from Amazon Connect • Bring Your Own Channel for CCaaS
`serverAuthMode`	**Field Type** ServerAuthMode (enumeration of type string) **Description** Deprecated in API 53.0. Server authentication mode. Set this value to `None`. Values are: • `None` • `OAuth` Applies to the following implementations: • Service Cloud Voice with Partner Telephony • Service Cloud Voice with Partner Telephony from Amazon Connect
`telephonySettingsComponent`	**Field Type** string **Description** The name of the Lightning Web Component (LWC) that is used to display additional agent settings in the Omni-Channel widget. This value is in the format `mynamespace:componentName`, where `mynamespace` is the namespace associated with the Service Cloud Voice package that was created, and `componentName` is the FQDN of the Lightning component. Available in API version 54.0 and later. Applies to the following implementation: • Service Cloud Voice with Partner Telephony
`unifiedRoutingSupported (Beta)`	**Field Type** boolean

Field Name	Description
	Description Indicates whether unified routing is supported (`true`) or not supported (`false`) for voice calls in voice channels. The default value is `false`. Once this value is set to `true`, it can't be changed to `false`. Available in API version 63.0 and later. Applies to the following implementation: • Service Cloud Voice with Partner Telephony
`universalCallRecordingAccessSupported`	**Field Type** boolean **Description** If set to `true`, Universal Call Recording Access is turned on. The default value is `false`. If this value is set to `false` and you want to turn on Universal Call Recording, set this value to `true` and configure the service_cloud_voice.RecordingMediaProvider interface in your Apex integration class. Available in API version 54.0 and later. Applies to the following implementations: • Service Cloud Voice with Partner Telephony • Service Cloud Voice with Partner Telephony from Amazon Connect
`userSyncingSupported`	**Field Type** boolean **Description** If set to `true`, supports automated user syncing whenever a user is added to or removed from a contact center. The default value is `false`. Available in API version 53.0 and later. Applies to the following implementations: • Service Cloud Voice with Partner Telephony • Service Cloud Voice with Partner Telephony from Amazon Connect • Bring Your Own Channel for CCaaS
`vendorType`	**Field Type** ConversationVendorType (enumeration of type string) **Description** The Service Cloud feature the partner vendor supports. Possible values are: • `Amazon_Connect` — For Service Cloud Voice with Amazon Connect. • `BringYourOwnChannelPartner` — For Bring Your Own Channel for Messaging. Available in API version 60.0 and later.

Field Name	Description
	• `BringYourOwnContactCenter` — For Bring Your Own Channel for Contact Center as a Service (CCaaS). Available in API version 60.0 and later. • `ServiceCloudVoicePartner` — For Service Cloud Voice with Partner Telephony or Service Cloud Voice with Partner Telephony from Amazon Connect. Available in API version 53.0 and later.

Declarative Metadata Sample Definition

The following is an example of a ConversationVendorInfo component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ConversationVendorInfo xmlns="http://soap.sforce.com/2006/04/metadata"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <einsteinConversationInsightsSupported>true</einsteinConversationInsightsSupported>
    <partnerContactCenterListSupported>true</partnerContactCenterListSupported>
    <namedCredentialSupported>true</namedCredentialSupported>
    <partnerTransferDestinationsSupported>true</partnerTransferDestinationsSupported>
    <agentSSOSupported>true</agentSSOSupported>
    <keyProvisioningSupported>true</keyProvisioningSupported>
    <universalCallRecordingAccessSupported>true</universalCallRecordingAccessSupported>
    <partnerPhoneNumbersSupported>true</partnerPhoneNumbersSupported>
    <queueManagementSupported>true</queueManagementSupported>
    <clientAuthMode>SSO</clientAuthMode>
    <connectorUrl>https://exampleconnectorurl.com</connectorUrl>
    <customConfig>exampleCustomConfig__c</customConfig>
    <customLoginUrl>testurl</customLoginUrl>
    <integrationClass>ExampleIntegrationImpl</integrationClass>
    <masterLabel>Example Partner Name</masterLabel>
    <developerName>exampledevname</developerName>
    <namedCredential>exampleNamedCredential</namedCredential>
    <userSyncingSupported>true</userSyncingSupported>
    <vendorType>BringYourOwnContactCenter</vendorType>
</ConversationVendorInfo>
```

The following is an example `package.xml` that references the previous definition.

```
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>ConversationVendorInfo</name>
    </types>
    <version>59.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ConvIntelligenceSignalRule

Represents the conversation intelligence signal rule. The rule triggers actions based on real-time intelligence signals from your telephony system or keywords mentioned by support reps or customers. The rule contains a set of conditions (subrules) and the filter logic used to evaluate those conditions to determine whether to trigger actions.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

ConvIntelligenceSignalRule components have the suffix `.ConvIntelligenceSignalRule` and are stored in the `ConvIntelligenceSignalRule` folder.

Version

ConvIntelligenceSignalRule components are available in API version 62.0 and later.

Special Access Rules

This type requires an add-on license for Service Cloud Voice for Amazon Connect, Service Cloud Voice for Partner Telephony with Amazon Connect, Service Cloud Voice for Partner Telephony, or Digital Engagement.

Fields

Field Name	Description
`actionType`	**Field Type** ConvIntelligenceActionType (enumeration of type string) **Description** Required. The conversation intelligence signal type. Values are: • `AlertSupervisor`–Sends an alert to the supervisor. • `AlertSupervisorAndAgent`–Sends an alert to the rep and supervisor. • `LaunchFlow`–Triggers an auto-launched flow. If set, also set `ActionValue`. • `LaunchNBA`–Recommends the next best action to the rep.
`actionValue`	**Field Type** string **Description** Action to perform based on the `actionType` specified. If `actionType` is set to LaunchFlow, this value is the `developerName` of the flow to be launched. For example, EmailAlert.

Field Name	Description
	For all other `actionType` values, don't set this parameter.
`active`	**Field Type** boolean **Description** Required. Indicates whether the conversation intelligence signal rule is active (`true`) or inactive (`false`). The default value is `false`.
`channelAddressIdentifier`	**Field Type** string **Description** Required. ID (`ChannelAddressIdentifier`) of the Messaging channel or name (`InternalName`) of the Voice channel.
`channelType`	**Field Type** string **Description** Required. Channel type. For Messaging, possible values are: • `AppleBusinessChat`—Represents Apple Messages for Business. • `Custom`—Represents Bring Your Own Channel for Messaging or Bring Your Own Channel for CCaaS. • `EmbeddedMessaging`—Represents Messaging for In-App and Web. • `Facebook` • `Text` • `WhatsApp` For Voice, set this parameter to `Phone`.
`criteria`	**Field Type** string **Description** Required. Filter logic applied to the rule conditions (subrules). For example, ((1 AND 2) OR 3). The numbers in the formula are derived from the `ConvIntelligenceSignalSubRule.order` value plus 1. For example, filter logic (1 AND 2) is calculated by adding the first condition (`order`=0) with the second condition (`order`=1).
`developerName`	**Field Type** string **Description** Requiired. API name of the conversation intelligence signal rule.

Field Name	Description
`participantRole`	**Field Type** ConvParticipantRole (enumeration of type string) **Description** If `service` is set to KeywordMatch, this value determines whether the rule applies to utterances made by reps, customers, or both roles. Possible values are: Possible values are: • `Agent` • `AgentOrCustomer` • `Customer` If `Service` is not set to KeywordMatch, don't set this parameter.
`ruleName`	**Field Type** string **Description** Required. Name of the conversation intelligence signal rule.
`service`	**Field Type** ConvIntelligenceService (enumeration of type string) **Description** Required. Salesforce- or partner-provided intelligence source. For Salesforce-provided intelligence sources, set this parameter to `KeywordMatch`. For partner-provided intelligence sources, possible values are: • `KeywordMatch` • `AmazonConnectContactLens` If none of the options apply to you, contact your Salesforce representative for the service name.
`subrule`	**Field Type** ConvIntelligenceSignalSubRule[] **Description** A set of intelligence rules used to measure an agent or customer's sentiment during a voice call.

ConvIntelligenceSignalSubRule

Represents a condition (subrule) within a conversation intelligence signal rule.

Field Name	Description
`operandValue`	**Field Type** string **Description** Required. Value of the signal type used to determine if the rule condition is met.
`operator`	**Field Type** ConvIntelligenceOperator (enumeration of type string) **Description** Required. Filter logic operator used to determine if the rule condition is met. Possible values are: • `Equals` • `GreaterThan` • `LessThan` • `NotEquals`
`order`	**Field Type** int **Description** Required. Order the condition appears in relation to the other conditions in the list, with zero (0) being the first condition listed. If `type` is set to Keyword, the maximum value is 24. For all other `type` values, the maximum value is 4. This value is used when applying filter logic to the rule.
`type`	**Field Type** ConvIntelligenceType (enumeration of type string) **Description** Required. Type of conversation intelligence signal used by the rule to determine whether to trigger an action. This value depends on the `ConvIntelligenceSignalRule.channelType` and `ConvIntelligenceSignalRule.service` values. If `service` is set to KeywordMatch, possible values are: • `Keyword`–A word or group of words spoken or typed. If `service` is set to `AmazonConnectContactLens`, possible values are: • `Category`–Category name defined in your telephony system. If `service` is set to another value, contact your Salesforce representative for the conversation intelligence signal types available for your intelligence source.

Declarative Metadata Sample Definition

The following is an example of a ConvIntelligenceSignalRule component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ConvIntelligenceSignalRule xmlns="http://soap.sforce.com/2006/04/metadata">
    <actionType>AlertSupervisor</actionType>
    <active>true</active>
<channelAddressIdentifier>a12bc345-1303-44c2-866c-f30d546b58de</channelAddressIdentifier>

    <channelType>Phone</channelType>
    <criteria>1 OR 2</criteria>
    <developerName>ConvIntelligenceRuleAPIName</developerName>
    <participantRole>AgentOrCustomer</participantRole>
    <ruleName>ConvIntelligenceRuleName</ruleName>
    <service>KeywordMatch</service>
    <subrule>
        <operandValue>escalate_level_1</operandValue>
        <operator>Equals</operator>
        <order>0</order>
        <type>Keyword</type>
    </subrule>
    <subrule>
        <operandValue>escalate_level_2</operandValue>
        <operator>Equals</operator>
        <order>1</order>
        <type>Keyword</type>
    </subrule>
</ConvIntelligenceSignalRule>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>ConversationIntelligenceSignalRule</name>
    </types>
    <version>62.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

CorsWhitelistOrigin

Represents an origin in the CORS allowlist.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. Because changing terms in our code can break current implementations, we maintained this metadata type's name.

File Suffix and Directory Location

CorsWhitelistOrigin components have the suffix `.corswhitelistorigin` and are stored in the `corswhitelistorigins` folder.

Version

CorsWhitelistOrigin components are available in API version 32.0 and later.

Fields

Field Name	Field Type	Description
`urlPattern`	String	A URL pattern for the origin. The origin URL pattern must include the HTTPS protocol and a domain name, and can include a port. The wildcard character (*) is supported and must be in front of a second-level domain name. For example, `https://*.example.com` adds all subdomains of `example.com` to the allowlist. Google Chrome™ and Mozilla® Firefox® browser extensions are also allowed as resources in API version 53 and later. Chrome extensions must use the prefix `chrome-extension://` and 32 characters without digits or capital letters, for example `chrome-extension://abdkkegmcbiomijcbdaodaflgehfffed`. Firefox extensions must use the prefix `moz-extension://` and an 8-4-4-4-12 format of small alphanumeric characters, for example `moz-extension://1234ab56-78c9-1df2-3efg-4567891hi1j2`. The origin URL pattern can be an IP address. But an IP address and a domain that resolve to the same address aren't the same origin, and you must add them to the CORS allowlist as separate entries.

Declarative Metadata Sample Definition

Here's an example package manifest used to deploy or retrieve the CorsWhitelistOrigin metadata for an organization.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>CorsWhitelistOrigin</name>
    </types>
    <version>32.0</version>
</Package>
```

Here's an example of a CorsWhitelistOrigin component.

```
<?xml version="1.0" encoding="UTF-8"?>
<CorsWhitelistOrigin xmlns="http://soap.sforce.com/2006/04/metadata">
```

```
    <developerName>CorsWhitelistEntry1</developerName>
    <urlPattern>https://*.example.com</urlPattern>
</CorsWhitelistOrigin>
```

Usage

CORS (cross-origin resource sharing) is a W3C recommendation that enables Web browsers to request resources from origins other than their own. For example, using CORS, a JavaScript script at `https://www.example.com` could request a resource from `https://www.salesforce.com`.

If a browser that supports CORS makes a request to an origin in your allowlist, Salesforce returns the origin in the `Access-Control-Allow-Origin` HTTP header, along with any additional CORS HTTP headers. If the origin isn't allow listed, Salesforce returns HTTP status code 404.

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

CspTrustedSite

Represents a trusted URL. For each CspTrustedSite component, you can specify Content Security Policy (CSP) directives and permissions policy directives. Each CSP directive allows Lightning components, third-party APIs, and WebSocket connections to access a resource type from the trusted URL. If the Permissions-Policy HTTP header is enabled, each permissions policy directive grants the trusted URL access to a browser feature. In API version 58.0 and earlier, CspTrustedSite components included only CSP directives and were referred to as CSP Trusted Sites.

This type extends the Metadata metadata type and inherits its `fullName` field.

Declarative Metadata File Suffix and Directory Location

CspTrustedSite components are stored in the `cspTrustedSites` directory of the corresponding package directory. The file name matches the unique name of the trusted site, and the extension is `.cspTrustedSite`.

Version

CspTrustedSite components are available in API version 39.0 and later.

Fields

Field	Field Type	Description
`canAccessCamera`	boolean	Indicates whether this CspTrustedSite can access the user's camera (`true`) or not (`false`). The default value is `false`. This field takes effect only when the `enablePermissionsPolicy` field equals `true` and

Field	Field Type	Description
		the `grantCameraAccess` field equals `TrustedUrls` in the SecuritySettings metadata API type. This field is available in API version 59.0 and later.
`canAccessMicrophone`	boolean	Indicates whether this CspTrustedSite can access the user's microphone (`true`) or not (`false`). The default value is `false`. This field takes effect only when the `enablePermissionsPolicy` field equals `true` and the `grantMicrophoneAccess` field equals `TrustedUrls` in the SecuritySettings metadata API type. This field is available in API version 59.0 and later.
`context`	CspTrustedSiteContext (enumeration of type string)	Declares the scope of the CSP directives for this trusted URL. • `All`—Apply the CSP directives to all supported context types. • `Communities`—Apply the CSP directives to Experience Builder sites only. • `FieldServiceMobileExtension`—Apply the CSP directives to the Field Service Mobile Extensions only. This value is available in API version 47.0 and later. • `LEX`—Apply the CSP directives to Lightning Experience pages only. • `VisualForce`—Apply the CSP directives to custom Visualforce pages only. This value is available in API version 55.0 and later. For custom Visualforce pages, content is restricted to trusted URLs only if the page's `cspHeader` attribute is set to `true`. This field is available in API version 44.0 and later.
`description`	string	The description of this trusted URL.
`endpointUrl`	string	Required. The URL for this CspTrustedSite. This field must include a domain name and can include a port. For example, `https://example.com` or `https://example.com:8080`. To reduce repetition, you can use the wildcard character `*` (asterisk). For example, `*.example.com`. For a third-party API, the URL must begin with https://. For example, `https://example.com`. For a WebSocket connection, the URL must begin with wss://. For example, `wss://example.com`. Otherwise, the URL cannot be malformed. Examples of malformed URLs that fail a syntax check are

Field	Field Type	Description
		`malformed^url.example.com`, and `https://{subdomain}.example.com`. To add an `EndpointUrl` based on parameters, build the URL before you add it to this Metadata Type.
`isActive`	boolean	Required. Indicates whether this CspTrustedSite is active (`true`) or not (`false`). The default value is `true`.
`isApplicableToConnectSrc`	boolean	Indicates whether Lightning components, third-party APIs, and WebSocket connections can load URLs using script interfaces from this trusted URL (`true`) or not (`false`). This field has a default value of `false`. This field is available in API version 48.0 and later.
`isApplicableToFontSrc`	boolean	Indicates whether Lightning components, third-party APIs, and WebSocket connections can load fonts from this trusted URL (`true`) or not (`false`). This field has a default value of `false`. This field is available in API version 48.0 and later.
`isApplicableToFrameSrc`	boolean	Indicates whether Lightning components, third-party APIs, and WebSocket connections can load resources contained in `<iframe>` elements from this trusted URL (`true`) or not (`false`). This field has a default value of `false`.This field is available in API version 48.0 and later.
`isApplicableToImgSrc`	boolean	Indicates whether Lightning components, third-party APIs, and WebSocket connections can load images from this trusted URL (`true`) or not (`false`). This field has a default value of `false`. This field is available in API version 48.0 and later.
`isApplicableToMediaSrc`	boolean	Indicates whether Lightning components, third-party APIs, and WebSocket connections can load audio and video from this trusted URL (`true`) or not (`false`). This field has a default value of `false`. In API version 59.0 and later, for each trusted URL, at least one CSPTrustedSite starting with `isApplicable` or `canAccess` must be set to `true.` In API version 50.0 to 58.0, if all `isApplicable` fields are `false`, the `isApplicableToImgSrc` field is set to `true`. In API version 49.0 and earlier, if all `isApplicable` fields are `false`, these fields all default to `true`. This field is available in API version 48.0 and later.
`isApplicableToStyleSrc`	boolean	Indicates whether Lightning components, third-party APIs, and WebSocket connections can load style sheets from this trusted URL (`true`) or not (`false`). This field has a default value of `false`. This field is available in API version 48.0 and later.

Field	Field Type	Description
mobileExtension	string	Reserved for future use.

Declarative Metadata Sample Definition

A sample XML definition of a trusted site is shown below.

```
<?xml version="1.0" encoding="UTF-8"?>
<CspTrustedSite xmlns="http://soap.sforce.com/2006/04/metadata">
    <canAccessCamera>false</canAccessCamera>
    <canAccessMicrophone>true</canAccessMicrophone>
    <description>Used for Lightning component callout to mapping web service</description>

    <context>LEX</context>
    <endpointUrl>https://www.maptestsite.net/</endpointUrl>
    <isActive>true</isActive>
    <isApplicableToConnectSrc>true</isApplicableToConnectSrc>
    <isApplicableToFontSrc>true</isApplicableToFontSrc>
    <isApplicableToFrameSrc>false</isApplicableToFrameSrc>
    <isApplicableToImgSrc>true</isApplicableToImgSrc>
    <isApplicableToMediaSrc>false</isApplicableToMediaSrc>
    <isApplicableToStyleSrc>true</isApplicableToStyleSrc>
</CspTrustedSite>
```

Usage

For each CSPTrustedSite component, at least one field starting with `grantAccess` or `isApplicableTo` must be set to `true`.

In API versions 50.0 to 58.0, if all `isApplicable` fields are `false`, the `isApplicableToImgSrc` field is set to `true`. In API version 49.0 and earlier, if all `isApplicable` fields are `false`, those fields all default to `true`.

To ensure smooth integration across Salesforce products, Salesforce includes URLs in each of the CSP directives that correspond to the `isApplicable` fields, even though those URLs aren't defined as CspTrustedSite components. Salesforce regularly updates those URLs based on the latest requirements.

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

CustomApplication

CustomApplication represents a custom or standard application. In API version 29.0 and earlier, CustomApplication represents only a custom application. An application is a list of tab references, with a description and a logo. This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

Custom and standard applications have the suffix `.app` and are stored in the `applications` folder.

Note: Retrieving a component of this metadata type in a project makes the component appear in any Profile and PermissionSet components that are retrieved in the same package.

Version

Custom applications are available in API version 10.0 and later. Standard applications are available in API version 30.0 and later.

Fields

Field Name	Field Type	Description
`actionOverrides`	AppActionOverride[]	Represents an action override for an application. Use it to create, update, edit, or delete action overrides. This field is available for Lightning Experience in API version 38.0 and later.
`brand`	AppBrand	The color scheme and logo used for the app. This field is available for Lightning Experience in API version 38.0 and later.
`consoleConfig`	ServiceCloudConsoleConfig	Represents configuration settings for a Salesforce console app. This field is available in API version 42.0 and later.
`defaultLandingTab`	string	The `fullName` of a standard tab or custom tab that opens when this application is selected.
`description`	string	The optional description text of the application.
`formFactors`	FormFactor (enumeration of type string)	Indicates the form factors for which the app is visible for Lightning Experience. Valid values are: • `Null` (no value)—For a desktop using Salesforce Classic • `Small`—For a mobile device using the Salesforce mobile app • `Medium`—Reserved for future use • `Large`—For a desktop using Lightning Experience This field is available in API version 38.0 and later. As of API version 38.0, `formFactors` is set to `Large` for existing Salesforce Classic apps, except for Salesforce Classic consoles. Salesforce Classic apps installed from packages created before API version 38.0 also have `formFactors` set to `Large`. For Salesforce Classic

Field Name	Field Type	Description
		apps in packages created with API 38.0 or later, you must set `formFactors` to `Large` for Salesforce Classic apps to appear in the Lightning Experience desktop. As of API version 47.0, the `Small` value is supported for Lightning apps. The `formFactors` field can be set to `Small` or `Large` for Lightning apps, and it can be set to `Null` or `Large` for Salesforce Classic apps.
`isNavAutoTempTabsDisabled`	boolean	Indicates whether the navigation automatically creates temporary tabs settings. Applies only to Lightning apps with standard navigation. Available in API version 43.0 and later.
`isNavPersonalizationDisabled`	boolean	Indicates whether navigation personalization is disabled. Applies only to Lightning apps. Available in API version 43.0 and later.
`isNavTabPersistenceDisabled`	boolean	Indicates whether workspace tabs are cleared for each new console session (`true`) or not (`false`). Applies only to Lightning apps with console navigation. Available in API version 54.0 and later.
`isServiceCloudConsole`	boolean	Indicates if the application is a Salesforce Classic console app. For Lightning Experience console apps, this field is `null` and the `navType` field is set to `Console`.
`label`	string	The name of the application.
`logo`	string	The optional reference to the image document for a Salesforce app or Salesforce console app.
`navType`	NavType (enumeration of type string)	Not updateable. Indicates the type of navigation the app uses. The value `Standard` is for a Lightning app with standard navigation. The value `Console` is for a Lightning app with console navigation. This field is available in API version 38.0 and later.
`preferences`	AppPreferences	Represents the preferences for a Salesforce Classic console app. All of the AppPreferences fields are required. This field is available in API version 42.0 and later.
`profileActionOverrides`	AppProfileActionOverride[]	A list of the Lightning Experience record page ProfileActionOverrides that are assigned to this custom app. When a user invokes the custom app, a matching ProfileActionOverride assignment takes precedence over existing overrides for the record page specified in ActionOverride. You can override a record page for the custom app by record type and profile.

Field Name	Field Type	Description
		In API version 45.0 and later, you can override a home page for the custom app by profile.
`setupExperience`	string	The type of Setup experience associated with the app. Valid values are: • `all`—Represents the full Setup tree. • `essentials`—Represents the Essentials Setup tree, which contains a subset of Setup items configured for Essentials edition. • `service`—Represents the Service Setup tree, which contains a subset of Setup items configured for Service Console. A `null` value is equivalent to `all`. Previous valid values `AllSetup`, `ServiceSetup`, and `EssentialsSetup` have been deprecated. This field is available in API version 39.0 and later.
`subscriberTabs`	string[]	Represents the list of tabs appended by a subscriber to a Lightning app installed from a managed package. Records in a subscriber tab always open as primary tabs. This field is available in API version 41.0 and later.
`tabs`	string[]	The list of tabs included in this application. In API version 12.0, the `fullName` for built-in tabs like Home, Account, and Reports, is the name of the tab (Home, for example). In API version 13.0 and later, built-in tabs are prefixed with `standard-`. For example, to reference the Account tab you would use `standard-Account`. In API version 42.0, this field was renamed from `tab` to `tabs`.
`uiType`	UiType (enumeration of type string)	Not updateable. Identifies the type of custom app. The value is: • `Aloha` for Salesforce Classic • `Lightning` for Lightning Experience This field is available in API version 38.0 and later.
`utilityBar`	string	The developer name of the utility bar associated with this app. We recommend assigning a utility bar to only one Lightning App, because utility bars are shared. Sharing means that if you change the utility bar in one app, it automatically changes in all apps associated with it.

Field Name	Field Type	Description
		This field is available in API version 38.0 and later.
`workspaceConfig`	AppWorkspaceConfig	Represents how records open in a Salesforce console app. Required if `isServiceCloudConsole` is `true`. In API version 42.0, this field was renamed to `workspaceConfig` from `workspaceMappings`.

AppActionOverride

Represents an action override for an application. Use it to create, update, edit, or delete action overrides. AppActionOverride inherits from ActionOverride and extends it by one field, `pageOrSobjectType`. Available for Lightning Experience in API version 38.0 and later.

Field Name	Field Type	Description
`actionName`	string	The only valid value is `view` for API version 43.0 and earlier. The value `tab` is supported for API version 44.0 and later.
`comment`	string	Any comments you want associated with the override.
`content`	string	Set this field if `type` is set to `flexipage`. It refers to the name of the page to use as the override. To reference installed components, use the format of ***`Component_namespace__Component_name`***.
`formFactor`	FormFactor(enumeration of type string)	The size of the page being overridden. If the `type` field is set to `flexipage`, set this field to `Large` to override the View action with a Lightning page in Lightning Experience. The `Large` value represents the Lightning Experience desktop environment and is valid only for the `flexipage` and `lightningcomponent` types. The `Small` value represents the Salesforce mobile app on a phone or tablet. The `Medium` value is reserved for future use. The `null` value (which is the same as specifying no value) represents Salesforce Classic. This field is available in API version 37.0 and later and is part of the feature for creating and editing record pages in Lightning Experience. Lightning component overrides return different `FormFactor` values depending on the API version used. • In API version 41.0 and earlier, Lightning component overrides return only the `null` value (no value), representing the Salesforce Classic environment. • In API version 42.0, if you specify different Lightning component overrides for Lightning Experience and mobile, one component is selected randomly for both overrides and its `FormFactor` value is returned. If there's a conflict between Lightning components, and a Visualforce page override is also specified for Salesforce Classic, the Visualforce page takes precedence.

Field Name	Field Type	Description
		• In API version 43.0 and later, a Lightning component override for Lightning Experience returns the `Large` value and a Lightning component override for mobile returns the `Small` value, as expected.
`pageOrSobjectType`	string	The name of the sObject type being overridden. Valid values are `standard` and `custom`. This value must be `standard-home` when actionName is `tab`.
`skipRecordTypeSelect`	boolean	Set this field to `true` if you prefer that any new records created by this action override aren't forwarded to the record type selection page. This field is only valid if the `actionName` is a "create" type (like `new`), and `type` is set to `visualforce`.
`type`	ActionOverrideType (enumeration of type string)	Required. Represents the type of action override. The valid values are `Flexipage` and `Default`. A `Flexipage` AppActionOverride set to App Default can't be deleted via Metadata API. Instead, remove the override using the page assignment wizard in the Lightning App Builder UI.

AppBrand

The color scheme and logo used for the app. Available for Lightning apps in API version 38.0 and later.

Field Name	Field Type	Description
`footerColor`	string	Optional. Determines the footer color in the app. Specify the color with a hexadecimal code, such as #0000FF for blue.
`headerColor`	string	Optional. Determines the header color in the app. Specify the color with a hexadecimal code, such as #0000FF for blue.
`logo`	string	The optional reference to the image document for the application.
`logoVersion`	int	An optional version number for the logo.
`shouldOverrideOrgTheme`	boolean	Indicates whether to override the global theme for the org. When `true`, the color scheme and logo that the user has set are used. When `false`, the global theme for the org is used, even if the user has set a color scheme and logo.

AppComponentList

Represents custom console components (Visualforce pages) assigned to a Salesforce console app. In API version 42.0, this type was renamed from CustomApplicationComponents to AppComponentList.

Field Name	Field Type	Description
`alignment`	string	Required. Determines how custom console components are aligned in the footer of a Salesforce console app.
`components`	string[]	The name of a custom console component assigned to a Salesforce console app. In API version 42.0, this field was renamed from `customApplicationComponent` to `components`.

AppPreferences

Represents the preferences for a Salesforce Classic console app. All of the AppPreferences fields are required. Available in API version 42.0 and later.

Field Name	Field Type	Description
`enableCustomizeMyTabs`	boolean	Indicates if a Salesforce Classic console app has Customize My Tabs enabled. If enabled, users can hide, display, and organize items in the navigation tab.
`enableKeyboardShortcuts`	boolean	Indicates if a Salesforce Classic console app has keyboard shortcuts enabled. Shortcuts let users perform actions by pressing a combination of keys instead of having to use a mouse. After keyboard shortcuts are enabled, several default shortcuts are available for customization. Before you can create custom shortcuts, a developer must define the shortcut's action with the `addEventListener()` method in the Salesforce Console Integration Toolkit. You can't create keyboard shortcuts for actions performed outside of the console. This field is required if `isServiceCloudConsole` is `true`.
`enableListViewHover`	boolean	Indicates if a Salesforce Classic console app has list view hovers enabled. If set to `true`, summary information is displayed about a record in a responsive list when the user hovers over a record name. For cases, hover over the subject field.
`enableListViewReskin`	boolean	Indicates if Salesforce Classic console apps use responsive list views instead of Salesforce Classic lists views.
`enableMultiMonitorComponents`	boolean	Indicates if a Salesforce Classic console app has multi-monitor components enabled, which lets users move portions of a console from their browsers to locations on their screens. This field is required if `isServiceCloudConsole` is `true`.
`enablePinTabs`	boolean	Indicates if a Salesforce Classic console app has pinned tabs enabled, which lets users pin primary tabs to the tab bar for quick access.
`enableTabHover`	boolean	Indicates if a Salesforce Classic console app has tab hover enabled. If enabled, summary information is displayed about a record in an overlay when the user hovers over a tab.

Field Name	Field Type	Description
`enableTabLimits`	boolean	Indicates whether limits are enabled on the number of primary tabs and subtabs that can be opened in a Salesforce Classic console session. When `true`, values for `tabLimitConfig` are required
`saveUserSessions`	boolean	Indicates if a Salesforce Classic console app saves user sessions automatically. If enabled, when console users close their browsers or log out of Salesforce, any previously open tabs display when users log in again. Required if `isServiceCloudConsole` is `true`.

AppProfileActionOverride

Represents a ProfileActionOverride for a custom app. This type inherits from ProfileActionOverride on page 1633 and extends it by one field, `profile`. Available for Lightning Experience in API version 39.0 and later. In API version 45.0 and later, you can override a home page for the custom app by profile.

Field Name	Field Type	Description
`actionName`	string	Required. The name of the action. The only valid values are `Tab` and `View`. If `pageOrSobjectType` is `record-home`, this field must be `View`. The `View` action is supported only when ProfileActionOverride is being specified as part of a CustomApplication. In API version 45.0 and later, this action is supported only when ProfileActionOverride is being specified as part of a CustomApplication, `pageOrSobjectType` is `standard-home`, and this field is `Tab`.
`content`	string	Read-only. Represents the name of the Lightning page being used as the override.
`formFactor`	FormFactor (enumeration of type string)	Required. The size of the page being overridden. The `Large` value represents the Lightning Experience desktop environment.
`pageOrSobjectType`	string	Required. The name of the page being overridden. The only valid values are `record-home` and `standard-home`. If the `actionName` is `Tab`, this field must be `standard-home`
`profile`	string	The profile associated with the ProfileActionOverride.
`recordType`	string	The record type associated with the override. If `pageOrSobjectType` is `standard-home`, this field must be `null`. This field is required when `actionName` is set to `View`.
`type`	ActionOverrideType (enumeration of type string)	Required. Read-only. The type of action override. The only valid value is `flexipage`.

AppWorkspaceConfig

Represents how records open in a Salesforce console app. Required if `isServiceCloudConsole` is `true`. Available for Salesforce Classic console apps in API version 25.0 and later. Available for Lightning console apps in API version 41.0 and later. In API version 42.0, this type was renamed from WorkspaceMappings to AppWorkspaceConfig.

Field Name	Field Type	Description
`mappings`	WorkspaceMappingSingle[]	Represents how records for a specific tab open in a Salesforce console app. Required for each tab specified in the CustomApplication. In API version 42.0, this field was renamed from `workspaceMapping` to `mappings`.

WorkspaceMapping

Represents how records for a specific tab open in a Salesforce console app. Required for each tab specified in the CustomApplication. Available in API version 25.0 and later for Salesforce Classic console apps. Available in API version 41.0 and later for Lightning console apps.

Field Name	Field Type	Description
`fieldName`	string	The name of the field that specifies the primary tab in which to display `tab` as a subtab. If not specified, `tab` opens as a primary tab.
`tab`	string	Required. Name of the tab.

CustomShortcut

Represents custom keyboard shortcuts assigned to a Salesforce console app in Salesforce Classic. Before you can create custom shortcuts, a developer must define the shortcut's action with the `addEventListener()` method in the Salesforce Console Integration Toolkit. You can't create keyboard shortcuts for actions performed outside of the console. Available in API version 28.0 and later.

Field Name	Field Type	Description
`action`	string	Required. The action performed in the console when a user presses the keyboard shortcut.
`active`	boolean	Required. Indicates whether the keyboard shortcut is active (`true`) or not (`false`).
`keyCommand`	string	Required. The combination of keys a user presses to trigger the keyboard shortcut. Keyboard shortcuts aren't case-sensitive, but they display as uppercase on setup pages in the Salesforce user interface so that they're easier to read. Each key command can include up to four modifier keys followed by one non-modifier key. Modifier and non-modifier keys are separated by the `+` key. Modifier keys can occur in any order, but you must place non-modifier keys at the end of the key command sequence. For example, `SHIFT+CTRL+ALT+META +A`.

Field Name	Field Type	Description
		Valid modifier keys are: • SHIFT • CTRL • ALT • META (represents the COMMAND key on Macs) Valid non-modifier keys are letters A through Z and numbers 0 through 9. Other valid keys are: • TAB • ENTER • PAUSE/BREAK • CAPS LOCK • ESC • SPACE • PAGE UP • PAGE DOWN • END • HOME • LEFT ARROW • UP ARROW • RIGHT ARROW • DOWN ARROW • PRINT SCREEN • INSERT • DELETE • RIGHT WINDOW • NUMPAD 0 • NUMPAD 1 • NUMPAD 2 • NUMPAD 3 • NUMPAD 4 • NUMPAD 5 • NUMPAD 6 • NUMPAD 7 • NUMPAD 8 • NUMPAD 9 • MULTIPLY • ADD • SUBTRACT

Field Name	Field Type	Description
		• DECIMAL POINT • DIVIDE • F1 • F2 • F3 • F4 • F5 • F6 • F7 • F8 • F9 • F10 • F11 • F12 • NUM LOCK • SCROLL LOCK • ; • = • , • – • . • / • ` • [•] • \ • '
description	string	The optional description text for the keyboard shortcut.
eventName	string	Required. Code available to developers who want to add custom shortcut functions to the console via the Salesforce Console Integration Toolkit.

DefaultShortcut

Represents default keyboard shortcuts assigned to a Salesforce console app. After you enable keyboard shortcuts for a console, several default shortcuts are available for customization. These include opening and closing tabs, moving between tabs, and saving records. Available in API version 28.0 and later.

Field Name	Field Type	Description
`action`	string	Required. The action performed in the console when a user presses the keyboard shortcut. Valid values are: • `FOCUS_CONSOLE` • `FOCUS_NAVIGATOR_TAB` • `FOCUS_DETAIL_VIEW` • `FOCUS_PRIMARY_TAB_PANEL` • `FOCUS_SUBTAB_PANEL` • `FOCUS_LIST_VIEW` • `FOCUS_FIRST_LIST_VIEW` • `FOCUS_SEARCH_INPUT` • `MOVE_LEFT` • `MOVE_RIGHT` • `UP_ARROW` • `DOWN_ARROW` • `OPEN_TAB_SCROLLER_MENU` • `OPEN_TAB` • `CLOSE_TAB` • `ENTER` • `EDIT` • `SAVE`
`active`	boolean	Required. Indicates whether the keyboard shortcut is active (`true`) or not (`false`).
`keyCommand`	string	Required. The combination of keys a user presses to trigger the keyboard shortcut. Keyboard shortcuts aren't case-sensitive, but they display as uppercase on setup pages in the Salesforce user interface so that they're easier to read. Each key command can include up to four modifier keys followed by one non-modifier key. Modifier and non-modifier keys are separated by the `+` key. Modifier keys can occur in any order, but you must place non-modifier keys at the end of the key command sequence. For example, `SHIFT+CTRL+ALT+META +A`. Valid modifier keys are: Valid non-modifier keys are letters A through Z and numbers 0 through 9. Other valid keys are:

KeyboardShortcuts

Represents keyboard shortcuts assigned to a Salesforce console app. Required if `isServiceCloudConsole` is `true`. Available in API version 28.0 and later.

Field Name	Field Type	Description
`customShortcuts`	CustomShortcut[]	Represents custom keyboard shortcuts assigned to a Salesforce console app in Salesforce Classic. Before you can create custom shortcuts, a developer must define the shortcut's action with the `addEventListener()` method in the Salesforce Console Integration Toolkit. You can't create keyboard shortcuts for actions performed outside of the console. In API version 42.0, this field was renamed from `customShortcut` to `customShortcuts`.
`defaultShortcuts`	DefaultShortcut[]	Represents default keyboard shortcuts assigned to a Salesforce console app. After you enable keyboard shortcuts for a console, several default shortcuts are available for customization. These include opening and closing tabs, moving between tabs, and saving records. In API version 42.0, this field was renamed from `defaultShortcut` to `defaultShortcuts`.

ListPlacement

Represents how lists display in a Salesforce console app. Required if `isServiceCloudConsole` is `true`. Available in API version 25.0 and later.

Field Name	Field Type	Description
`height`	int	Height of the list in pixels or percentage. Required if `location` is top.
`location`	string	Required. Location of the list on the screen. Valid values are: • full • top • left
`units`	string	Required. Represents if `height` or `width` is in pixels or percentage.
`width`	int	Width of the list in pixels or percentage. Required if `location` is left.

LiveAgentConfig

Represents your organization's settings for using Chat in the Salesforce Console.

Field Name	Field Type	Description
`enableLiveChat`	boolean	Specifies whether Chat is enabled in your organization (`true`) or not (`false`).
`openNewAccountSubtab`	boolean	Specifies whether to open a new Account subtab in a Salesforce console app automatically (`true`) or not (`false`) when an agent accepts a chat.

Field Name	Field Type	Description
`openNewCaseSubtab`	boolean	Specifies whether to open a new Case subtab in a Salesforce console app automatically (`true`) or not (`false`) when an agent accepts a chat.
`openNewContactSubtab`	boolean	Specifies whether to open a new Contact subtab in a Salesforce console app automatically (`true`) or not (`false`) when an agent accepts a chat.
`openNewLeadSubtab`	boolean	Specifies whether to open a new Lead subtab in a Salesforce console app automatically (`true`) or not (`false`) when an agent accepts a chat.
`openNewVFPageSubtab`	boolean	Specifies whether to open a new Visualforce page as a subtab in a Salesforce console app automatically (`true`) or not (`false`) when an agent accepts a chat.
`pageNamesToOpen`	string [array of strings]	Specifies the Visualforce pages to open in subtabs when an agent accepts a chat in a Salesforce console app. This field is available in API version 42.0 and later.
`showKnowledgeArticles`	boolean	Specifies whether to display the Knowledge component while using Chat in a Salesforce console app (`true`) or not (`false`).

PushNotification

Represents a set of push notifications, which are visual indicators on lists and detail pages that show when a record or field has changed during a user's session. Available for use if `isServiceCloudConsole` is `true`. Available in API version 28.0 and later.

Field Name	Field Type	Description
`fieldNames`	string]	The name of the field or fields that trigger push notifications for the selected object.
`objectName`	string	Required. Name of the object that triggers push notifications.

ServiceCloudConsoleConfig

Represents configuration settings for a Salesforce console app. Available in API version 42.0 and later.

Field Name	Field Type	Description
`componentList`	AppComponentList	Represents custom console components (Visualforce pages) assigned to a Salesforce console app.
`detailPageRefreshMethod`	string	Determines how detail pages refresh in a Salesforce console app. Required if `isServiceCloudConsole` is `true`. The valid values are: • `none` • `autoRefresh` • `flag`

Field Name	Field Type	Description
`footerColor`	string	Determines the footer color in a Salesforce console app. Specify the color with a hexadecimal code, such as #0000FF for blue.
`headerColor`	string	Determines the header color in a Salesforce console app. Specify the color with a hexadecimal code, such as #0000FF for blue.
`keyboardShortcuts`	KeyboardShortcuts	Represents the keyboard shortcuts for a Salesforce console app. Keyboard shortcuts let users perform actions by pressing a combination of keys instead of having to use a mouse.
`listPlacement`	ListPlacement	Represents how lists display in a Salesforce console app. Required if `isServiceCloudConsole` is `true`.
`listRefreshMethod`	string	Determines how lists refresh in a Salesforce console app. Required if `isServiceCloudConsole` is `true`. The valid values are: • `none` • `refreshList` • `refreshListRows`
`liveAgentConfig`	LiveAgentConfig	Represents the configurations for using Chat in the Salesforce Console.
`primaryTabColor`	string	Determines the primary tab color in a Salesforce console app. Specify the color with a hexadecimal code, such as #0000FF for blue.
`pushNotifications`	PushNotification[]	Represents push notifications for a Salesforce console app. Push notifications are visual indicators on lists and detail pages that show when a record or field has changed during a user's session. For example, assume that two support agents are working on the same case. If one agent changes the `Priority`, a push notification displays to the other agent so the agent notices the change and doesn't duplicate the effort.
`tabLimitConfig`	TabLimitConfig	Represents the maximum number of primary tabs and subtabs allowed in one Salesforce console session. Required if `enableTabLimits` is `true`.
`whiteListedDomains`	string[]	Any external domains that users can access from within a Salesforce console app. For example, `www.yourdomain.com`.

TabLimitConfig

Represents the maximum number of primary tabs and subtabs allowed in one Salesforce console session. Required if `enableTabLimits` is `true`. Available in API version 36.0 and later.

Field Name	Field Type	Description
`maxNumberOfPrimaryTabs`	string	The maximum number of primary tabs allowed in one console session. Valid values are: • 5 • 10

Field Name	Field Type	Description
		• 20 • 30
maxNumberOfSubTabs	string	The maximum number of subtabs allowed in one console session. Valid values are: • 5 • 10 • 15

Usage

You can't delete custom app ProfileActionOverrides by deploying with `destructiveChange.xml`. To delete a ProfileActionOverride, retrieve the app. In the app definition file, find the `<profileActionOverrides>` section, and remove the `<content>` row. Then, change the `<type>` value in that same section to `default` instead of `flexipage`. Do this for every override you want to reset. After making the changes, rezip the folder and deploy.

You can remove one override at a time each with its own deploy, or you can remove multiple overrides in a single deploy. However, we recommend that you do a fresh retrieve every time you want to delete a new override. Don't use a previously retrieved file.

Retrieving Apps

To retrieve apps in your organization, use the CustomApplication type name in the `package.xml` manifest file. You can either retrieve all apps or specify which apps to retrieve in the types section of `package.xml`.

To retrieve all apps in your organization—custom and standard apps, specify the wildcard character (`*`), as follows.

```
<types>
    <members>*</members>
    <name>CustomApplication</name>
</types>
```

Note: In API version 29.0 and earlier, use of the wildcard returns only all custom applications but not standard applications.

To retrieve a custom app, specify the app name.

```
<types>
    <members>MyCustomApp</members>
    <name>CustomApplication</name>
</types>
```

To retrieve a standard app, add the `standard__` prefix to the app name. For example, to retrieve the Chatter standard app, specify `standard__Chatter`.

```
<types>
    <members>standard__Chatter</members>
    <name>CustomApplication</name>
</types>
```

To retrieve an app that is part of an installed package, add the package namespace prefix followed by two underscores and the app name. For example, if the package namespace is `myInstalledPackageNS` and the app name is `PackageApp`, specify `myInstalledPackageNS__PackageApp`, as follows.

```
<types>
    <members>myInstalledPackageNS__PackageApp</members>
    <name>CustomApplication</name>
</types>
```

Declarative Metadata Sample Definition

Here's the definition of a custom Lightning Experience app:

```
<?xml version="1.0" encoding="UTF-8"?>
<CustomApplication xmlns="http://soap.sforce.com/2006/04/metadata">
    <actionOverrides>
        <actionName>View</actionName>
        <comment>Action override created by Lightning App Builder during
activation.</comment>
        <content>Custom_Mobile_Oppty_Page</content>
        <formFactor>Small</formFactor>
        <skipRecordTypeSelect>false</skipRecordTypeSelect>
        <type>Flexipage</type>
        <pageOrSobjectType>Opportunity</pageOrSobjectType>
    </actionOverrides>
    <actionOverrides>
        <actionName>View</actionName>
        <comment>Action override created by Lightning App Builder during
activation.</comment>
        <content>Custom_Mobile_Oppty_Page</content>
        <formFactor>Large</formFactor>
        <skipRecordTypeSelect>false</skipRecordTypeSelect>
        <type>Flexipage</type>
        <pageOrSobjectType>Opportunity</pageOrSobjectType>
    </actionOverrides>
    <brand>
        <headerColor>#EE1518</headerColor>
        <shouldOverrideOrgTheme>true</shouldOverrideOrgTheme>
    </brand>
    <description>Manage inventory and deliveries for our warehouses.</description>
    <formFactors>Small</formFactors>
    <formFactors>Large</formFactors>
    <isNavAutoTempTabsDisabled>false</isNavAutoTempTabsDisabled>
    <isNavPersonalizationDisabled>false</isNavPersonalizationDisabled>
    <label>Warehouse Lightning</label>
    <navType>Standard</navType>
    <profileActionOverrides>
        <actionName>View</actionName>
        <content>Warehouse_test_page</content>
        <formFactor>Large</formFactor>
        <pageOrSobjectType>Warehouse__c</pageOrSobjectType>
        <type>Flexipage</type>
        <profile>Admin</profile>
    </profileActionOverrides>
```

```
    <profileActionOverrides>
        <actionName>View</actionName>
        <content>Warehouse_test_page</content>
        <formFactor>Small</formFactor>
        <pageOrSobjectType>Warehouse__c</pageOrSobjectType>
        <type>Flexipage</type>
        <profile>Admin</profile>
    </profileActionOverrides>
    <setupExperience>all</setupExperience>
    <tabs>standard-Feed</tabs>
    <tabs>standard-File</tabs>
    <tabs>standard-Account</tabs>
    <tabs>standard-Case</tabs>
    <tabs>Merchandise__c</tabs>
    <tabs>Invoice__c</tabs>
    <tabs>Warehouse__c</tabs>
    <tabs>Delivery__c</tabs>
    <tabs>standard-report</tabs>
    <tabs>standard-Dashboard</tabs>
    <uiType>Lightning</uiType>
</CustomApplication>
```

The following is a definition of a standard app (Chatter):

```
<?xml version="1.0" encoding="UTF-8"?>
<CustomApplication xmlns="http://soap.sforce.com/2006/04/metadata">
    <defaultLandingTab>standard-home</defaultLandingTab>
    <label>Collaboration</label>
    <tabs>standard-Chatter</tabs>
    <tabs>standard-UserProfile</tabs>
    <tabs>standard-OtherUserProfile</tabs>
    <tabs>standard-CollaborationGroup</tabs>
    <tabs>standard-File</tabs>
</CustomApplication>
```

Declarative Metadata Sample Definition—Salesforce Console

The following is the definition of a custom app where `isServiceCloudConsole` is `true`:

```
<?xml version="1.0" encoding="UTF-8"?>
<CustomApplication xmlns="http://soap.sforce.com/2006/04/metadata">
    <consoleConfig>
        <componentList>
            <alignment>left</alignment>
            <components>MyComponent</components>
        </componentList>
        <detailPageRefreshMethod>autoRefresh</detailPageRefreshMethod>
        <keyboardShortcuts>
            <customShortcuts>
                <action>MyCustomShortcutAction</action>
                <active>true</active>
                <keyCommand>X</keyCommand>
                <description>Custom Shortcut example</description>
                <eventName>myCustomShortcutExample</eventName>
```

```
    </customShortcuts>
    <defaultShortcuts>
        <action>FOCUS_CONSOLE</action>
        <active>true</active>
        <keyCommand>ESC</keyCommand>
    </defaultShortcuts>
    <defaultShortcuts>
        <action>FOCUS_NAVIGATOR_TAB</action>
        <active>true</active>
        <keyCommand>V</keyCommand>
    </defaultShortcuts>
    <defaultShortcuts>
        <action>FOCUS_DETAIL_VIEW</action>
        <active>true</active>
        <keyCommand>SHIFT+S</keyCommand>
    </defaultShortcuts>
    <defaultShortcuts>
        <action>FOCUS_PRIMARY_TAB_PANEL</action>
        <active>true</active>
        <keyCommand>P</keyCommand>
    </defaultShortcuts>
    <defaultShortcuts>
        <action>FOCUS_SUBTAB_PANEL</action>
        <active>true</active>
        <keyCommand>S</keyCommand>
    </defaultShortcuts>
    <defaultShortcuts>
        <action>FOCUS_LIST_VIEW</action>
        <active>true</active>
        <keyCommand>N</keyCommand>
    </defaultShortcuts>
    <defaultShortcuts>
        <action>FOCUS_FIRST_LIST_VIEW</action>
        <active>true</active>
        <keyCommand>SHIFT+F</keyCommand>
    </defaultShortcuts>
    <defaultShortcuts>
        <action>FOCUS_SEARCH_INPUT</action>
        <active>true</active>
        <keyCommand>R</keyCommand>
    </defaultShortcuts>
    <defaultShortcuts>
        <action>MOVE_LEFT</action>
        <active>true</active>
        <keyCommand>LEFT ARROW</keyCommand>
    </defaultShortcuts>
    <defaultShortcuts>
        <action>MOVE_RIGHT</action>
        <active>true</active>
        <keyCommand>RIGHT ARROW</keyCommand>
    </defaultShortcuts>
    <defaultShortcuts>
        <action>UP_ARROW</action>
        <active>true</active>
```

```
            <keyCommand>UP ARROW</keyCommand>
        </defaultShortcuts>
        <defaultShortcuts>
            <action>DOWN_ARROW</action>
            <active>true</active>
            <keyCommand>DOWN ARROW</keyCommand>
        </defaultShortcuts>
        <defaultShortcuts>
            <action>OPEN_TAB_SCROLLER_MENU</action>
            <active>true</active>
            <keyCommand>D</keyCommand>
        </defaultShortcuts>
        <defaultShortcuts>
            <action>OPEN_TAB</action>
            <active>true</active>
            <keyCommand>T</keyCommand>
        </defaultShortcuts>
        <defaultShortcuts>
            <action>CLOSE_TAB</action>
            <active>true</active>
            <keyCommand>C</keyCommand>
        </defaultShortcuts>
        <defaultShortcuts>
            <action>ENTER</action>
            <active>true</active>
            <keyCommand>ENTER</keyCommand>
        </defaultShortcuts>
        <defaultShortcuts>
            <action>EDIT</action>
            <active>true</active>
            <keyCommand>E</keyCommand>
        </defaultShortcuts>
        <defaultShortcuts>
            <action>SAVE</action>
            <active>true</active>
            <keyCommand>CTRL+S</keyCommand>
        </defaultShortcuts>
    </keyboardShortcuts>
    <listPlacement>
        <location>left</location>
        <units>percent</units>
        <width>20</width>
    </listPlacement>
    <listRefreshMethod>refreshList</listRefreshMethod>
    <pushNotifications>
        <fieldNames>CreatedBy</fieldNames>
        <objectName>Campaign</objectName>
    </pushNotifications>
    <pushNotifications>
        <fieldNames>CustomField1__c</fieldNames>
        <objectName>CustomObject1__c</objectName>
    </pushNotifications>
</consoleConfig>
<defaultLandingTab>standard-home</defaultLandingTab>
```

```
    <isServiceCloudConsole>true</isServiceCloudConsole>
    <label>MyConsole</label>
    <preferences>
        <enableCustomizeMyTabs>false</enableCustomizeMyTabs>
        <enableKeyboardShortcuts>true</enableKeyboardShortcuts>
        <enableListViewHover>true</enableListViewHover>
        <enableListViewReskin>true</enableListViewReskin>
        <enableMultiMonitorComponents>true</enableMultiMonitorComponents>
        <enablePinTabs>true</enablePinTabs>
        <enableTabHover>false</enableTabHover>
        <enableTabLimits>false</enableTabLimits>
        <saveUserSessions>false</saveUserSessions>
    </preferences>
    <tabs>standard-Case</tabs>
    <tabs>standard-Account</tabs>
    <tabs>standard-Contact</tabs>
    <tabs>standard-Contract</tabs>
    <workspaceConfig>
        <mappings>
            <tab>standard-Case</tab>
        </mappings>
        <mappings>
            <fieldName>ParentId</fieldName>
            <tab>standard-Account</tab>
        </mappings>
        <mappings>
            <fieldName>AccountId</fieldName>
            <tab>standard-Contact</tab>
        </mappings>
        <mappings>
            <tab>standard-Contract</tab>
        </mappings>
    </workspaceConfig>
</CustomApplication>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SEE ALSO:

CustomTab

CustomApplicationComponent

Represents a custom console component (Visualforce page) assigned to a CustomApplication that is marked as a Salesforce console. Custom console components extend the capabilities of Salesforce console apps. See Customize a Console with Custom Components in Salesforce Classic in Salesforce Help.

File Suffix and Directory Location

Custom application components have the suffix `.customApplicationComponent` and are stored in the `customApplicationComponents` folder.

Version

Custom applications are available in API version 25.0 and later.

Fields

Field Name	Field Type	Description
`buttonIconUrl`	string	The address of a page that hosts an icon for the button.
`buttonStyle`	string	The inline style used to define how the button looks.
`buttonText`	string	The label on the button used to launch the custom console component.
`buttonWidth`	int	The pixel width of the button displayed in the Salesforce console.
`height`	int	The pixel height of the window used to display the custom console component.
`isHeightFixed`	boolean	Required. Indicates whether users can change the custom console component height (`false`) or not (`true`).
`isHidden`	boolean	Required. Indicates whether the custom console component is hidden from users (`true`) or not (`false`).
`isWidthFixed`	boolean	Required. Indicates whether users can change the component width (`false`) or not (`true`).
`visualforcePage`	string	Required. Name of the Visualforce page that represents the custom console component.
`width`	int	The pixel width of the window used to display the custom console component.

Declarative Metadata Sample Definition

The following is the definition of a custom application component:

```
<?xml version="1.0" encoding="UTF-8"?>
<CustomApplicationComponent xmlns="http://soap.sforce.com/2006/04/metadata">
    <buttonIconUrl>https://salesforce.com</buttonIconUrl>
    <buttonStyle>buttonStyleCSS</buttonStyle>
    <buttonText>buttonText</buttonText>
    <buttonWidth>200</buttonWidth>
    <height>200</height>
    <isHeightFixed>false</isHeightFixed>
    <isHidden>false</isHidden>
```

```
    <isWidthFixed>false</isWidthFixed>
    <visualforcePage>MyVisualforcePage</visualforcePage>
    <width>50</width>
</CustomApplicationComponent>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

CustomFeedFilter

Represents a custom feed filter that limits the feed view to feeds from the Cases object. The custom feed filter shows only feed items that satisfy the criteria specified in the CustomFeedFilter definition. This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

CustomFeedFilter components have the suffix `.feedFilter` and are stored in the `feedFilters` folder.

Version

CustomFeedFilter components are available in API version 35.0 and later.

Fields

Field Name	Field Type	Description
`criteria`	FeedFilterCriterion on page 645 []	The criterion that defines which feed items are shown when the filter is applied. The feed filter displays all feed items that satisfy the criteria.
`description`	string	The description of the custom feed filter. For example, specify what feed items that filter shows.
`label`	string	Required. The API label of the custom feed filter.
`isProtected`	boolean	An auto-generated value. It currently has no impact.

FeedFilterCriterion

Represents the conditions that a feed item must satisfy to be displayed when a feed filter is applied.

Field Name	Field Type	Description
`feedItemType`	FeedItemType (enumeration of type string)	Required. The type of feed items that the filter shows. The feed item type can be one of the following values:

Field Name	Field Type	Description
		• AttachArticleEvent • CallLogPost • CanvasPost • CaseCommentPost • ChangeStatusPost • ChatTranscriptPost • ContentPost • CreateRecordEvent • EmailMessageEvent • LinkPost • MilestoneEvent • QuestionPost • PollPost • ReplyPost • SocialPost • TextPost
`feedItemVisibility`	FeedItemVisibility (enumeration of type string)	The visibility of feed items that the filter shows. For example, you can show only poll posts that are visible internally. Valid values are: • AllUsers • InternalUsers
`relatedSObjectType`	string	The API name of the object that the feed item refers to. This field is typically used with the CreateRecordEvent feed item type. For example, a feed filter can show CreateRecordEvent feed items for the Cases object.

Declarative Metadata Sample Definition

The following is an example of a CustomFeedFilter on page 645 component.

```
<?xml version="1.0" encoding="UTF-8"?>
<CustomFeedFilter xmlns="http://soap.sforce.com/2006/04/metadata">
    <criteria>
        <feedItemType>CreateRecordEvent</feedItemType>
        <relatedSObjectType>MyCO01__c</relatedSObjectType>
    </criteria>
    <criteria>
        <feedItemType>CreateRecordEvent</feedItemType>
```

```
        <relatedSObjectType>Case</relatedSObjectType>
    </criteria>
    <criteria>
        <feedItemType>PollPost</feedItemType>
        <feedItemVisibility>InternalUsers</feedItemVisibility>
    </criteria>
    <label>Sample Custom Feed Filter</label>
</CustomFeedFilter>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>myCaseFeedFilter</members>
        <name>CustomFeedFilter</name>
    </types>
    <version>64.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

CustomFieldDisplay

Represents the view type assigned to product attribute custom fields. This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

CustomFieldDisplay components have the suffix `.customFieldDisplay`.

Version

CustomFieldDisplay components are available in API version 63.0 and later.

Fields

Field Name	Field Type	Description
`displayType`	CustomFieldDisplayType (enumeration of type string)	Required. The view type of the product attribute custom fields. Values are: • `ColorSwatch`

Field Name	Field Type	Description
		• `Dropdown` • `Pill`
`fieldApiName`	string	Required. The unique name of the product attribute, for example, color_c.
`isProtected`	boolean	Optional. An auto-generated value that doesn't impact the behavior of the metadata type. The default value is `false`.
`masterLabel`	string	Required. The primary label for this object.

Declarative Metadata Sample Definition

The following is an example of a CustomFieldDisplay component.

```
<?xml version="1.0" encoding="UTF-8"?>
<CustomFieldDisplay xmlns="http://soap.sforce.com/2006/04/metadata">
 <masterLabel>cfd1</masterLabel>
 <fieldApiName>Color__c</fieldApiName>
 <displayType>Pill</displayType>
 <isProtected>false</isProtected>
</CustomFieldDisplay>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
 <types>
 <members>*</members>
 <name>CustomFieldDisplay</name>
 </types>
 <version>63.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

CustomHelpMenuSection

Represents the section of the Lightning Experience help menu that the admin added to display custom, org-specific help resources for the org. The custom section contains help resources added by the admin. This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

CustomHelpMenuSection components have the suffix `.customHelpMenuSection` and are stored in the `customHelpMenuSections` folder.

Version

CustomHelpMenuSection components are available in API version 45.0 and later.

Fields

Field Name	Field Type	Description
`customHelpMenuItems`	CustomHelpMenuItems[]	Items included in the custom section. Specify up to 15 items.
`masterLabel`	string	Required. Name of the custom section. Only one custom section can be added to the Lightning Experience help menu. Specify up to 80 characters.

CustomHelpMenuItems

Items included in the custom section. Specify up to 15 items.

Field Name	Field Type	Description
`linkURL`	string	Required. The URL for the resource.
`masterLabel`	string	Required. The name of the resource. Specify up to 100 characters.
`sortOrder`	int	Required. The order of the item within the custom section. Valid values are `1` through `15`.

Declarative Metadata Sample Definition

The following is an example of a CustomHelpMenuSection component.

```
<?xml version="1.0" encoding="UTF-8"?>
<CustomHelpMenuSection xmlns="http://soap.sforce.com/2006/04/metadata">
    <masterLabel>MyOrgCustomHelp</masterLabel>
    <customHelpMenuItems>
        <linkUrl>https://www.yourcompanyhelp.com/gettingstarted</linkUrl>
        <masterLabel>Getting Started</masterLabel>
        <sortOrder>1</sortOrder>
    </customHelpMenuItems>
    <customHelpMenuItems>
        <linkUrl>https://www.yourcompanyhelp.com/features</linkUrl>
        <masterLabel>Feature to Start Using Right Away</masterLabel>
        <sortOrder>2</sortOrder>
    </customHelpMenuItems>
    <customHelpMenuItems>
```

```
        <linkUrl>https://www.yourcompanyhelp.com/salestips</linkUrl>
        <masterLabel>Tips for Sales Team Members</masterLabel>
        <sortOrder>3</sortOrder>
    </customHelpMenuItems>
</CustomHelpMenuSection>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>MyOrgCustomHelp</members>
        <name>CustomHelpMenuSection</name>
    </types>
    <version>45.0</version>
</Package>
```

CustomIndex

Represents an index used to increase the speed of queries.This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

CustomIndex components have the suffix .indx-meta and are stored in the `customindex`folder.

Version

CustomIndex is available in API versions 50.0 and later.

Special Access Rules

To use this metadata and create a custom index, review Indexes in *Best Practices for Deployments with Large Data Volumes*, and then contact Salesforce Customer Support.

Fields

Field Name	Field Type	Description
`allowNullValues`	boolean	Indicates whether null values are allowed in the index (`true`) or not (`false`). The default value is `false`.

Declarative Metadata Sample Definition

The following is an example of a CustomIndex component.

```
<?xml version="1.0" encoding="UTF-8" ?>
<CustomIndex xmlns="http://soap.sforce.com/2006/04/metadata">
```

```
    <allowNullValues>false</allowNullValues>
</CustomIndex>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

CustomLabels

The CustomLabels metadata type allows you to create custom labels that can be localized for use in different languages, countries, and currencies.

This type extends the Metadata metadata type and inherits its `fullName` field. Custom labels are custom text values, up to 1,000 characters in length that can be accessed from Apex classes or Visualforce pages. For more information, see "Custom Labels" in Salesforce Help.

Declarative Metadata File Suffix and Directory Location

Master custom label values are stored in the `CustomLabels.labels` file. Translations for custom labels can be retrieved through Translations in Metadata API. Translations are stored in files under the `translations` folder with the name format of *`localeCode`*`.translation`, where *`localeCode`* is the locale code of the translation language. The supported locale codes are listed in Language on page 2233.

Version

CustomLabels components are available in API version 14.0 and later.

Fields

Field	Field Type	Description
`fullName`	string	Required. The name of the custom label bundle. Inherited from Metadata, this field is defined in the WSDL for this metadata type. It must be specified when creating, updating, or deleting. See `createMetadata()` to see an example of this field specified for a call.
`labels`	CustomLabel[]	A list of custom labels.

CustomLabel

This metadata type represents a custom label. This type extends the Metadata metadata type and inherits its `fullName` field.

Field	Field Type	Description
`categories`	string	A comma-separated list of categories for the label. This field can be used in filter criteria when creating custom label list views. Maximum of 255 characters.
`fullName`	string	Required. The name of the custom label. Inherited from Metadata, this field is defined in the WSDL for this metadata type. It must be specified when creating, updating, or deleting. See `createMetadata()` to see an example of this field specified for a call.
`language`	string	Required. The language of the translated custom label.
`protected`	boolean	Required. Indicates whether this component is protected (`true`) or not (`false`). Protected components can't be linked to or referenced by components created in the installing organization.
`shortDescription`	string	Required. An easily recognizable term to identify this custom label. This description is used in merge fields.
`value`	string	Required. The translated custom label. Maximum of 1000 characters.

Usage

Use CustomLabels with the wildcard character (*) for members in the `package.xml` manifest file to retrieve all custom labels that are defined in your organization. CustomLabels doesn't support retrieving one or more custom labels by name. To retrieve specific labels by name, use CustomLabel and specify the label names as members.

Declarative Metadata Sample Definition

This is a sample XML definition of a custom label component.

```
<?xml version="1.0" encoding="UTF-8"?>
<CustomLabels xmlns="http://soap.sforce.com/2006/04/metadata">
    <labels>
        <fullName>quoteManual</fullName>
        <value>This is a manual quote.</value>
        <language>en_US</language>
        <protected>false</protected>
        <shortDescription>Manual Quote</shortDescription>
    </labels>
    <labels>
        <fullName>quoteAuto</fullName>
        <value>This is an automatically generated quote.</value>
        <language>en_US</language>
        <protected>false</protected>
        <shortDescription>Automatic Quote</shortDescription>
    </labels>
</CustomLabels>
```

This is a sample manifest file for retrieving all custom labels in the organization by using the CustomLabels type.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <fullName>MyPkg</fullName>
    <types>
      <members>*</members>
      <name>CustomLabels</name>
    </types>
    <version>64.0</version>
</Package>
```

This is a sample manifest file for retrieving two custom labels by name. Notice it uses the CustomLabel singular type.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <fullName>MyPkg</fullName>
    <types>
      <members>quoteManual</members>
      <members>quoteAuto</members>
      <name>CustomLabel</name>
    </types>
    <version>64.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

CustomLabels Limitation

Before you use the CustomLabels metadata type, understand the limitations of this feature. You can't retrieve the CustomLabels metadata type with a namespace.

SEE ALSO:

Translations

Custom Metadata Types (CustomObject)

Represents the metadata associated with a custom metadata type.

For more information, see Custom Metadata Types.

File Suffix and Directory Location

A custom metadata type is defined as a custom object and is stored in the objects folder. Custom metadata types have a suffix of `__mdt` (instead of `__c` for custom objects). Custom metadata type field names have a suffix of `__c`, like other custom fields. Custom metadata type field names must be dot-qualified with the name of the custom metadata type to which they belong.

Names of custom metadata types must be unique within their namespace. All custom metadata types belong to the `CustomMetadata` namespace and can optionally belong to a second namespace. In your organization, you can use custom metadata types with your namespace and also other organizations' namespaces.

Version

Custom metadata type components are available in API version 31.0 and later.

Special Access Rules

To create custom metadata types, you must have the "Author Apex" permission. Apex code can create, read, and update (but not delete) custom metadata records, as long as the metadata is subscriber-controlled and visible from within the code's namespace. You can edit records in memory but not upsert or delete them. Apex code can deploy custom metadata records, but not via a DML operation. Moreover, DML operations aren't allowed on custom metadata in the Partner or Enterprise APIs. Customers who install a managed custom metadata type can't add new custom fields to it. With unpackaged metadata, both developer-controlled and subscriber-controlled access behave the same: like subscriber-controlled access. Refer to Trust, but Verify: Apex Metadata API and Security to learn more.

Note: Audit fields (`CreatedDate`, `CreatedBy`, `LastModifiedDate`, `LastModifiedBy`, `SystemModStamp`) remain uneditable.

Fields

Custom metadata types can contain the following CustomObject fields.

To make the fields on your custom metadata types unique and indexable, mark your fields as `Unique` and `ExternalId`.

Field Name	Field Type	Description
`description`	string	A description of the custom metadata type. This field can contain a maximum of 1,000 characters.
`fields`	CustomField[]	Represents one or more custom fields in the custom metadata type.
`gender`	Gender	Indicates the gender of the noun that represents the object. This field is used for languages where words need different treatment depending on their gender.
`label`	string	A label that represents the object throughout the Salesforce Setup user interface. Custom metadata types are visible only through the recently used objects list on the Lightning Platform Home Page and in the packaging user interface.
`pluralLabel`	string	The plural version of the label value.
`startsWith`	StartsWith (enumeration of type string)	Indicates whether the noun starts with a vowel, a consonant, or a special character. This field is used for languages where words need different treatment depending on their first character.

Field Name	Field Type	Description
`visibility`	SetupObjectVisibility (enumeration of type string)	This field returns the visibility of a custom metadata type. The following values are valid: • `Public`—If the custom setting or custom metadata type is packaged, it's accessible to all subscribing organizations. • `Protected`—If the custom object, custom setting, or custom metadata type is in a managed package, it's accessible only to the developer org. Subscribing orgs can't access it. • `PackageProtected`—If the custom metadata type is `PackageProtected`, it's only accessible by the custom Apex code in the package. Use this value to secure secrets such as API access keys and security tokens. Available in API version 47.0 and later. The default value is `Public`.

Declarative Metadata Sample Definition

In this example, Picklists R Us creates its Reusable Picklist custom metadata type by deploying a file in the objects folder, named `ReusablePicklistOption__mdt.object`, with these contents.

```
<?xml version="1.0" encoding="UTF-8"?>
<CustomObject xmlns="http://soap.sforce.com/2006/04/metadata">
   <fields>
      <fullName>AlphaSort__c</fullName>
      <defaultValue>false</defaultValue>
      <externalId>false</externalId>
      <label>Sorted Alphabetically</label>
      <type>Checkbox</type>
   </fields>
   <label>Reusable Picklist</label>
   <pluralLabel>Reusable Picklist</pluralLabel>
   <visibility>Public</visibility>
</CustomObject>
```

This excerpt from a `package.xml` file shows the use of dot notation and the `__mdt` suffix. If you're using a namespace, for example `picklist1234`, the full name of `ReusablePicklistOption__mdt` would be `picklist1234__ReusablePicklistOption__mdt`.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
...
   <types>
      <members>PicklistTest__c.PicklistTestField__c</members>
      <members>ReusablePicklistOption__mdt.Picklist__c</members>
      <members>ReusablePicklistOption__mdt.SortOrder__c</members>
      <members>PicklistUsage__mdt.Field__c</members>
      <members>PicklistUsage__mdt.Picklist__c</members>
```

```
      <members>PicklistUsage__mdt.SObjectType__c</members>
      <members>ReusablePicklist__mdt.AlphaSort__c</members>
      <name>CustomField</name>
   </types>
...
   <types>
      <members>PicklistTest__c</members>
      <members>ReusablePicklistOption__mdt</members>
      <members>PicklistUsage__mdt</members>
      <members>ReusablePicklist__mdt</members>
      <name>CustomObject</name>
   </types>
...
   <version>64.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

1. CustomMetadata
   Represents a record of a custom metadata type.

CustomMetadata

Represents a record of a custom metadata type.

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

CustomMetadata components have the suffix `.md` and are stored in the `customMetadata` folder. Unlike custom metadata types, custom metadata records don't have a double-underscore suffix. Custom metadata record names are prepended with their custom metadata type name, excluding the `__mdt` suffix but including the namespace of any types in an installed managed package.

Version

CustomMetadata components are available in API version 31.0 and later.

Special Access Rules

To create custom metadata records, you must have the "Customize Application" permission.

Fields

Field Name	Field Type	Description
`description`	string	A description of the custom metadata record. This field can contain a maximum of 1,000 characters.
`label`	string	A label that represents the object throughout the Salesforce Setup user interface. Custom metadata records are currently visible only through the packaging user interface.
`protected`	boolean	Boolean. Indicates whether the record is protected (true) or not (false). When a custom metadata type is released in a managed package, access is limited in specific ways. • Code that's in the same managed package as custom metadata records can read the records. • Code that's in the same managed package as custom metadata types can read the records that belong to that type. • Code that's in a managed package that doesn't contain either the type or the protected record can't read the protected records. • Code that the subscriber creates and code that's in an unmanaged package can't read the protected records. • The developer can modify protected records with a package upgrade or by using the Metadata Apex classes (if the Apex code is in the same namespace as either the records or their type). The subscriber can't read or modify protected records. The developer name of a protected record can't be changed after release. • The subscriber can't create records of a protected type. Records that are hidden by these access rules are also unavailable to REST, SOAP, SOQL, and Setup.
`values`	CustomMetadataValue[]	Represents one or more values for custom fields on the custom metadata record.

CustomMetadataValue

Represents a value for a custom field on the custom metadata record.

Field Name	Field Type	Description
`field`	string	Required. The non-object-qualified name of a custom field in the custom metadata type. This value corresponds to the name of a field on the custom metadata record's custom metadata type. Include the namespace (if the type is from a managed package) and the `__c` suffix. The name of the custom metadata type isn't required. For example, `picklist1234__AlphaSort__c`.
`value`	Any type	The value on a custom metadata record. Where fields are EntityDefinition and FieldDefinition, the qualified API names of the entity and the field it points to. This value can be null.

Declarative Metadata Sample Definitions

The following is an example of a CustomMetadata component. In this example, the sample app TravelApp deploys a Planets picklist, specifies its sort order, and adds picklist items to it.

Assuming Picklists R Us's namespace is `picklist1234`, to define the `Planets` picklist, TravelApp deploys a file in the `customMetadata` folder, named `picklist1234__ReusablePicklist.Planets.md`, with these contents. The `xsi:type` attribute specifies the type for the value of the `AlphaSort__c` checkbox field.

```
<?xml version="1.0" encoding="UTF-8"?>
<CustomMetadata xmlns="http://soap.sforce.com/2006/04/metadata"
                  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <description>All the planets in the solar system. Does not
                include asteroids.</description>
    <label>Planets</label>
    <values>
        <field>picklist1234__AlphaSort__c</field>
        <value xsi:type="xsd:boolean">false</value>
    </values>
</CustomMetadata>
```

Picklists R Us creates its Reusable Picklist Option custom metadata type by deploying a file in the objects folder, named `ReusablePicklist__mdt.object`, with these contents.

```
<?xml version="1.0" encoding="UTF-8"?>
<CustomObject xmlns="http://soap.sforce.com/2006/04/metadata">
    <fields>
        <fullName>Picklist__c</fullName>
        <externalId>false</externalId>
        <label>Picklist</label>
        <length>40</length>
        <required>true</required>
        <type>Text</type>
        <unique>false</unique>
    </fields>
    <fields>
        <fullName>SortOrder__c</fullName>
```

```
        <externalId>false</externalId>
        <label>Non-Alphabetical Sort Order</label>
        <precision>3</precision>
        <scale>0</scale>
        <required>false</required>
        <type>Number</type>
        <unique>false</unique>
    </fields>
    <label>Reusable Picklist Option</label>
    <pluralLabel>Reusable Picklist Options</pluralLabel>
</CustomObject>
```

To define the `Mars` picklist item, TravelApp deploys a file, named `picklist1234__ReusablePicklistOption.Mars.md`, with these contents. This component file specifies types that apply to the `ReusablePicklistOption__mdt` custom fields.

```
<?xml version="1.0" encoding="UTF-8"?>
<CustomMetadata xmlns="http://soap.sforce.com/2006/04/metadata"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <label>Mars</label>
    <values>
        <field>picklist1234__Picklist__c</field>
        <value xsi:type="xsd:string">Planets</value>
    </values>
    <values>
        <field>picklist1234__SortOrder__c</field>
        <value xsi:type="xsd:int">4</value>
    </values>
</CustomMetadata>
```

To define the `Motel6` picklist item, TravelApp deploys a file, named `picklist1234__ReusablePicklistOption.Motel6.md`, with these contents.

```
<?xml version="1.0" encoding="UTF-8"?>
<CustomMetadata xmlns="http://soap.sforce.com/2006/04/metadata"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <label>Motel 6</label>
    <values>
        <field>picklist1234__Picklist__c</field>
        <value xsi:type="xsd:string">Hotels</value>
    </values>
</CustomMetadata>
```

Because the `SortOrder__c` field isn't required, this file doesn't require a value for `SortOrder__c`. Alternatively, the file could have explicitly specified a value with `xsi:nil` to ensure that `SortOrder__c` was cleared of any previous value.

```
<?xml version="1.0" encoding="UTF-8"?>
<CustomMetadata xmlns="http://soap.sforce.com/2006/04/metadata"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <label>Motel 6</label>
    <values>
        <field>picklist1234__Picklist__c</field>
        <value xsi:type="xsd:string">Hotels</value>
```

```
    </values>
    <values>
        <field>picklist1234__SortOrder__c</field>
        <value xsi:nil="true" />
    </values>
</CustomMetadata>
```

This excerpt from a `package.xml` file illustrates the inclusion of custom metadata types and their namespaces in custom metadata records' names. Assume that Picklists R Us's namespace is `picklist1234`.

```
<?xml version="1.0" encoding="UTF-8"?>
<package xmlns="http://soap.sforce.com/2006/04/metadata">
…
   <types>
      <members>picklist1234__ReusablePicklist.Hotels</members>
      <members>picklist1234__ReusablePicklist.Planets</members>
      <members>picklist1234__ReusablePicklistOption.Bellagio</members>
      <members>picklist1234__ReusablePicklistOption.Motel6</members>
      <members>picklist1234__ReusablePicklistOption.Mercury</members>
      <members>picklist1234__ReusablePicklistOption.Venus</members>
      <members>picklist1234__ReusablePicklistOption.Earth</members>
      <members>picklist1234__PicklistUsage.BookedHotel</members>
      <members>
         picklist1234__PicklistUsage.DestinationPlanetPL
      </members>
      <members>picklist1234__PicklistUsage.PlanetVisitedPl</members>
      <name>CustomMetadata</name>
   </types>
…
</package>
```

TravelApp, Inc.'s `package.xml` file uses a wildcard to install custom metadata, as is shown in this excerpt from their `package.xml` file. Unless you want to deploy or retrieve specific records, using a wildcard is easier than listing all of your custom metadata records in your `package.xml` file.

```
<types>
   <members>*</members>
   <name>CustomMetadata</name>
</types>
```

If the custom metadata is from a managed package, the name after the dot in the `package.xml` file—between the two dots in the file name—is qualified by the managed package's namespace. For example, assuming TravelApp uses the namespace `travelApp1234`, the first member element in the TravelApp `package.xml` file appears to Galactic Tours as:

```
<members>picklist1234__ReusablePicklist.travelApp1234__Hotels</members>
```

Here's another example. In this case, we have an instance of custom metadata record, whose EntityDefinition field points to a custom object named `SalesAgreement__c`. The FieldDefinition field points to the custom field `CustomerReference__c` on `SalesAgreement__c`. You can deploy new custom metadata records and retrieve existing ones with EntityDefinition and FieldDefinition fields using qualified API names of custom and standard entities and their fields.

```
<?xml version="1.0" encoding="UTF-8"?><values>
<field>EntityDefintionField__c</field>
<value xsi:type="xsd:string">v1__SalesAgreement__c</value>
</values>
```

```
<values>
<field>FieldDefinitionField__c</field>
<value xsi:type="xsd:string">v1__CustomerReference__c</value>
</values>
```

Usage

When specifying the `value` field in the CustomMetadataValue subtype, specify an appropriately typed object that's based on your field type definition. In declarative metadata definitions for CustomMetadataValue, use the `xsi:type` attribute of the value element. For example, to specify a boolean value: `<value xsi:type="xsd:boolean">true</value>`. Valid `xsi:type` attributes are:

Custom metadata value	Custom field definition
`xsi:type="xsd:boolean"`	Checkbox
`xsi:type="xsd:date"`	Date
`xsi:type="xsd:dateTime"`	Date/Time
`xsi:type="xsd:picklist"`	Picklist
`xsi:type="xsd:string"`	Text
`xsi:type="xsd:string"`	Phone
`xsi:type="xsd:string"`	TextArea
`xsi:type="xsd:string"`	URL
`xsi:type="xsd:string"`	Email
`xsi:type="xsd:int"`	Number/Percent, with scale equal to 0
`xsi:type="xsd:double"`	Number/Percent, with scale not equal to 0

You can also omit the `xsi:type` attribute. For example, `<value>true</value>`.

Although this attribute must be specified for any CustomMetadataValue, you can use an element with the `xsi:nil` attribute set to `true` to explicitly set the field's value to `null`. For example, `<value xsi:nil="true"/>`.

Using `null` field values differs from leaving out the CustomMetadataValue for a particular field entirely. If you leave out the CustomMetadataValue, the value of the field doesn't change. The field's value is `null` for newly deployed custom metadata records and left at its previous value for updated custom metadata records.

When you retrieve CustomMetadataValue objects, the `value` field of the returned object holds a value of the correct type, specified by `xsi:type` in the case of declarative metadata definitions.

Custom number fields are stored as double values. When you retrieve a value from a Number type field with a scale 0, you will see a decimal number. For example, if the value in UI is 1234567, a query through the API returns 1234567.0.

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

CustomNotificationType

Represents the metadata associated with a custom notification type.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

For more information about custom notifications, see Custom Notification Actions. This type extends the Metadata metadata type and inherits its `fullName` field.

Declarative Metadata File Suffix and Directory Location

The file suffix is `.notiftype` for the notification type definition. Notification types are stored in the `notificationtypes` directory of the corresponding package directory.

Version

CustomNotificationType components are available in API version 46.0 and later.

Fields

Field Name	Field Type	Description
`customNotifTypeName`	string	Required. Specifies a notification type name. Maximum number of characters: 80.
`description`	string	Specifies a general description of the notification type, which is displayed with the notification type name. Maximum number of characters: 255.
`desktop`	boolean	Required. Indicates whether the desktop delivery channel is enabled (`true`) or not (`false`).
`masterLabel`	string	Required. Specifies the label for the notification type.
`mobile`	boolean	Required. Indicates whether the mobile delivery channel is enabled (`true`) or not (`false`).
`slack`	boolean	Reserved for future use.

Declarative Metadata Sample Definition

The following is a definition of a custom notification type that is enabled for desktop and mobile.

```
<CustomNotificationType xmlns="http://soap.sforce.com/2006/04/metadata">
    <customNotifTypeName>Custom Notification</customNotifTypeName>
    <desktop>true</desktop>
    <masterLabel>Custom Notification</masterLabel>
    <mobile>true</mobile>
</CustomNotificationType>
```

Wildcard Support in the Manifest File

This metadata type doesn't support the wildcard character * (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

CustomObject

Represents a custom object that stores data unique to your org or an external object that maps to data stored outside your org.

This type extends the Metadata metadata type and inherits its `fullName` field.

Specify all relevant fields when you create or update a custom object. You can't update a single field on the object. For more information about custom objects, see Store Information That's Unique to Your Organization in Salesforce Help.

You can also use this metadata type to work with customizations of standard objects, such as accounts. For an example, see the section on Standard Objects in Sample package.xml Manifest Files in the *Metadata API Developer Guide*

All metadata components have a `fullName` field, which must be fully specified for any custom object.

For example, the following are fully specified names for a standard object and a custom object respectively:

```
Account
MyCustomObject__c
```

And the following is a fully specified name for an external object:

```
MyExternalObject__x
```

For sample Java code that creates a custom object, see Step 3: Walk Through the Java Sample Code on page 16.

Declarative Metadata File Suffix and Directory Location

Custom object names are automatically appended with __c. The file suffix is `.object` for the custom object or standard object file.

External object names are automatically appended with __x. The file suffix is `.object` for the external object file.

Custom, standard, and external objects are stored in the `objects` folder in the corresponding package directory.

Note: Retrieving a component of this metadata type in a project makes the component appear in any Profile and PermissionSet components that are retrieved in the same package.

Version

Custom objects are available in API version 10.0 and later. External objects are available in API version 32.0 and later.

Fields

Unless otherwise noted, all fields are creatable, filterable, and nillable.

Field Name	Field Type	Description
`actionOverrides`	ActionOverride[]	A list of action overrides on the object. This field is available in API version 18.0 and later.

Field Name	Field Type	Description
`allowInChatterGroups`	boolean	Indicates whether records of this custom object type can be added to Chatter groups. This field is available in API version 34.0 and later.
`businessProcesses`	BusinessProcess[]	A list of business processes associated with the object. This field is available in API version 17.0 and later.
`compactLayoutAssignment`	string	The compact layout assigned to the object. This field is available in API version 29.0 and later. This field is available for external objects in API version 42.0 and later.
`compactLayouts`	CompactLayout[]	A list of compact layouts associated with the object. This field is available in API version 29.0 and later. This field is available for external objects in API version 42.0 and later.
`customHelp`	string	The s-control that contains the help content if the object has customized help content. This field is available in API version 14.0 and later.
`customHelpPage`	string	The Visualforce page that contains the help content if the object has customized help content. This field is available in API version 16.0 and later.
`customSettingsType`	CustomSettingsType (enumeration of type string)	When this field is present, this component isn't a custom object, but a custom setting. This field returns the type of custom setting. The following string values are valid: • `List`—static data stored in cache, accessed as part of your application, and available org-wide. • `Hierarchy`—static data stored in cache, accessed as part of your application, and available based on a hierarchy of user, profile, or org. This value is the default. This field is available in API version 17.0 and later.
`customSettingsVisibility`	CustomSettingsVisibility (enumeration of type string)	When this field is present, this component isn't a custom object, but a custom setting. This field returns the visibility of the custom setting. The following string values are valid: • `Public`—if the custom setting is packaged, it's accessible to all subscribing orgs. • `Protected`—if the custom setting is in a managed package, it's accessible only to the developer org. Subscribing orgs can't access it. This value is the default. This field is available in API versions 17.0 through 33.0. In versions 34.0 and later, use the `visibility` field instead of this field.

Field Name	Field Type	Description
dataStewardGroup	string	Removed in API version 47.0.
dataStewardUser	string	Removed in API version 47.0.
deploymentStatus	DeploymentStatus (enumeration of type string)	Indicates the deployment status of the object.
deprecated	boolean	Reserved for future use.
description	string	A description of the object. Maximum of 1000 characters.
enableActivities	boolean	Indicates whether the object is enabled for activities (true) or not (false). Not available for external objects.
enableBulkApi	boolean	When enabled, the object is classified as an Enterprise Application object for usage tracking. When enabled, enableSharing and enableStreamingApi must also be enabled. This field is available in API version 31.0 and later.
enableDivisions	boolean	Indicates whether the object is enabled for divisions (true) or not (false). See Division in the *Salesforce Object Reference*. .
enableEnhancedLookup	boolean	Indicates whether the object is enabled for enhanced lookups (true) or not (false). In API version 28.0 and later, this field can also be used for the Account, Contact, and User objects. Enhanced lookups provide an updated lookup dialog interface that lets users filter, sort, and page through search results and customize search result columns. For more information about enhanced lookups, see "Enable Enhanced Lookups" in Salesforce Help.
enableFeeds	boolean	Indicates whether the object is enabled for feed tracking (true) or not (false). For more information, see "Customize Chatter Feed Tracking" in Salesforce Help. This field is available in API version 18.0 and later.
enableHistory	boolean	Indicates whether the object is enabled for history tracking (true) or not (false). Also available for standard objects in API version 29.0 and later. History tracking on the Account object includes person account history tracking.
enableLicensing	boolean	Indicates whether this object is licensed by Salesforce and users require a permission set license for it (true) or not (false). This field is available in API version 45.0 and later.

Field Name	Field Type	Description
`enableReports`	boolean	Indicates whether the object is enabled for reports (`true`) or not (`false`). Support for external objects is available in API version 38.0 and later.
`enableSearch`	boolean	Indicates whether the object's records can be found via SOSL and Salesforce searches. Corresponds to `Allow Search` in the user interface. By default, search is disabled for new custom objects. This field is available for custom objects in API version 35.0 and later. By default, search is disabled for new external objects. However, you can validate and sync an external data source to automatically create external objects. Syncing always enables search on the external object when search is enabled on the external data source, and vice versa.This field is available for external objects in API version 37.0 and later.
`enableSharing`	boolean	When enabled, the object is classified as an Enterprise Application object for usage tracking. When enabled, `enableBulkApi` and `enableStreamingApi` must also be enabled. This field is available in API version 31.0 and later.
`enableStreamingApi`	boolean	When enabled, the object is classified as an Enterprise Application object for usage tracking. When enabled, `enableBulkApi` and `enableSharing` must also be enabled. This field is available in API version 31.0 and later.
`eventType`	PlatformEventType (enumeration of type string)	This field applies only to platform events. Indicates the event type. The values are: • `HighVolume`—For a high-volume platform event. • `StandardVolume`—Deprecated. Creating a platform event with this event type is supported and returns an error. This field is available in API version 41.0 and later.
`externalDataSource`	string	Required and available for external objects only. The name of the external data source that stores the data for the external object. The data source is represented by the ExternalDataSource component. This field is available in API version 32.0 and later.

Field Name	Field Type	Description
`externalName`	string	Required and available for external objects only. The name of the table in the external data source that contains the data for the external object. This field is available in API version 32.0 and later.
`externalRepository`	string	Available for Salesforce Connect external objects only. Corresponds to `Display URL Reference Field` in the user interface. The external object's `Display URL` standard field values are automatically generated from the external system. For example, with the OData 2.0 adapter for Salesforce Connect, the value is based on the `link href` that's defined on the OData producer. You can override the default values with the values of a custom field on the same external object. Select the field name, and make sure that the custom field's values are valid URLs. This field is available in API version 32.0 and later.
`externalSharingModel`	SharingModel (enumeration of type string)	Indicates the external org-wide defaults for the object, which determines the access level for external users. This field is available in API version 31.0 and later.
`fields`	CustomField[]	Represents one or more fields in the object.
`fieldSets`	FieldSet	Defines the field set that exists on this object.
`fullName`	string	Inherited from Metadata, this field is defined in the WSDL for this metadata type. It must be specified when creating, updating, or deleting. See `createMetadata()` to see an example of this field specified for a call. This value can't be `null`.
`gender`	Gender	Indicates the gender of the noun that represents the object. This is used for languages where words need different treatment depending on their gender.
`household`	boolean	This field supports relationship groups, a feature available only with Salesforce for Wealth Management. For more information, see "Salesforce for Wealth Management" in Salesforce Help.
`historyRetentionPolicy`	HistoryRetentionPolicy	Reserved for future use.
`indexes`	Index[]	Defines the index for a custom big object.

Field Name	Field Type	Description
`label`	string	Label that represents the object throughout the Salesforce user interface. We recommend that you make object labels unique across all standard, custom, and external objects in the org.
`listViews`	ListView[]	Represents one or more *list views* associated with the object.
`namedFilter`	NamedFilter[]	Represents the metadata associated with a lookup filter. This metadata type is used to create, update, or delete lookup filter definitions. This component has been removed as of API version 30.0 and is only available in previous API versions. The metadata associated with a lookup filter is now represented by the lookupFilter field in the CustomField component. This field is available in API version 17.0 and later. This field has been removed as of API version 30.0 and is only available in prior versions. The metadata associated with a lookup filter is now represented by the lookupFilter field in the CustomField component.
`nameField`	CustomField	Required for custom objects. On external objects, the name field can instead be specified by setting `isNameField` to `true` in the CustomField component. The field that this object's name is stored in. Every custom object must have a name, usually a string or autonumber. Identifier for the custom object record. This name appears in page layouts, related lists, lookup dialogs, search results, and key lists on tab home pages. By default, this field is added to the custom object page layout as a required field.
`pluralLabel`	string	Plural version of the `label` value. Custom objects require a plural version of the label to ensure that object names are localizable.
`profileSearchLayouts`	ProfileSearchLayouts	Represents a user profile's search results layouts for an object. With profile-specific layouts, each user profile can have a different search results layout for an object. Available in API version 47.0 and later.
`publishBehavior`	PlatformEventPublishBehavior (enumeration of type string)	This field applies only to platform events. Indicates when platform event messages are published in a Lightning Platform transaction. This field applies to event messages published through the Lightning Platform, such as Apex, Process Builder, and Flow Builder, but not through Salesforce APIs. Valid values are:

Field Name	Field Type	Description
		• `PublishAfterCommit`—The event message is published only after a transaction commits successfully. If the transaction fails, the event message isn't published. • `PublishImmediately`—The event message is published when the publish call executes, regardless of whether the transaction succeeds. If you don't specify this field, the default value used is `PublishImmediately`. This field is available in API version 46.0 and later.
`recordTypes`	RecordType[]	An array of one or more record types defined for this object.
`recordTypeTrackFeedHistory`	boolean	Indicates whether the record type is enabled for feed tracking (`true`) or not (`false`). To set this field to `true`, the `enableFeeds` field on the associated CustomObject must also be `true`. For more information, see "Customize Chatter Feed Tracking" in Salesforce Help. This field is available in API version 19.0 and later.
`recordTypeTrackHistory`	boolean	Indicates whether history tracking is enabled for this record type (`true`) or not (`false`). To set `recordTypeTrackHistory` to true, the `enableHistory` field on the associated custom object must also be `true`. This field is available in API version 19.0 and later.
`searchLayouts`	SearchLayouts	The *Search Layouts* related list information for the object.
`sharingModel`	SharingModel(enumeration of type string)	Indicates the org-wide defaults for the object. Note: Using API version 29.0 and earlier, this field is read-only and can't be set using the Metadata API; you must use the Salesforce user interface. Using API version 30.0 and later, you can set this field for internal users using the API and the Salesforce user interface.
`sharingReasons`	SharingReason[]	The reasons why the object is being shared.
`sharingRecalculations`	SharingRecalculation[]	A list of custom sharing recalculations associated with the object.
`startsWith`	StartsWith (enumeration of type string)	Indicates whether the noun starts with a vowel, consonant, or is a special character. This is used for languages where words need different treatment depending on the first character. Valid values are listed in StartsWith.
`validationRules`	ValidationRule[]	An array of one or more validation rules on the object.

Field Name	Field Type	Description
`visibility`	SetupObjectVisibility (enumeration of type string)	This field returns the visibility of the custom object, custom setting, or custom metadata type. The following values are valid. • `Public`—If the custom object, custom setting, or custom metadata type is packaged, it's accessible to all subscribing orgs. • `Protected`—If the custom object, custom setting, or custom metadata type is in a managed package, it's accessible only to the developer org. Subscribing orgs can't access it. • `PackageProtected`— (Custom metadata type only) If the custom metadata type is `PackageProtected`, it's only accessible by the custom Apex code in the package. Use this value to secure secrets such as API access keys and security tokens. Available in API version 47.0 and later. The default value is `Public`. This field is available in API version 34.0 and later. For custom settings, this field replaces the `customSettingsVisibility` field.
`webLinks`	WebLink[]	An array of one or more weblinks defined for the object.

MktDataModelAttributes

This type is a Data Cloud subtype of CustomObject.

Field Name	Field Type	Description
`creationType`	DefinitionCreationType enumeration	Indicates how this object is added. Valid values availble in API version 62.0 and later are: • `Activation_Audience` • `Ad_Audience_Insights` • `ADG` • `Calculated_Insight` • `CG_Audience` • `Chunk` • `Directory_Table` • `External` • `Problem_Records` • `Segment_Membership` • `Semantic`

Field Name	Field Type	Description
		• `Transform` • `Vector_Embedding`
`dataModelTaxonomy`	string	When the model is a Standard Data Cloud model, a Reference to the Data Model from which this Object was started. Currently only supports the following strings: if the creationType is Standard, it must be Reference, if creationType is Custom, it must be View.
`description`	string	A description of the object. This field can contain a maximum of 521 characters. This field is available in API version 55.0 and later.
`isEnabled`	boolean	True indicates that the Data Model Object is enabled.
`isSegmentable`	boolean	True indicates that the Data Model Object can be used as a target for segmentation.
`isUsedForMetrics`	boolean	Indicates whether the Data Model Object is used for metrics (`true`) or not (`false`). This field is used to include additional attributes on the objects that are not present in the Data Model Object POJO. This field is available in API version 55.0 and later.
`objectCategory`	string	Reference to the Object Category. For modeling, the value is Profile, Engagement, or Other.
`referenceEntityGroup`	string	When this is a Standard Object, the Entity Group of the Object from the Reference Model.
`referenceEntityName`	string	When this is a Standard Object, the Name of the Object from the Reference Model.
`referenceEntitySubjectArea`	string	When this is a Standard Object, the Subject Area of the Object from the Reference Model.

MktDataLakeAttributes

Represents how Data Cloud receives the data. MktDataLakeAttributes is a Data Cloud subtype of CustomObject. Its components are available in API version 50.0 and later.

Special Access Rules

You need an org with Data Cloud license to access this object.

Field Name	Description
`creationType`	**Field Type** DefinitionCreationType enumeration of type string **Description** Indicates how this object is added. Values are:

Field Name	Description
	• Activation_Audience • Bridge • Curated • Custom • Derived • Ml_Prediction • Segment_Membership • Standard • System Valid values availble in API version 62.0 and later are: • Activation_Audience • Ad_Audience_Insights • ADG • Calculated_Insight • CG_Audience • Chunk • Directory_Table • External • Problem_Records • Segment_Membership • Semantic • Transform • Vector_Embedding
isEnabled	**Field Type** boolean **Description** Indicates whether the Landing Object is enabled.
objectCategory	**Field Type** string **Description** Reference to the Object Category. For landing object, these would be Profile, Behavioral, Other.

Declarative Metadata Additional Components

CustomObject definitions can include additional components defined in the custom object for declarative metadata. The following components are defined in the CustomObject:

- ActionOverride
- BusinessProcess
- CompactLayout
- CustomField
- FieldSet
- HistoryRetentionPolicy
- ListView
- RecordType
- SearchLayouts
- SharingReason
- SharingRecalculation
- ValidationRule
- WebLink

Declarative Metadata Sample Definition

```
<?xml version="1.0" encoding="UTF-8"?>
<CustomObject xmlns="http://soap.sforce.com/2006/04/metadata">
    <deploymentStatus>Deployed</deploymentStatus>
    <description>test object with one field for eclipse ide testing</description>
    <fields>
        <fullName>Comments__c</fullName>
        <description>add your comments about this object here</description>
       <inlineHelpText>This field contains comments made about this object</inlineHelpText>

        <label>Comments</label>
        <length>32000</length>
        <type>LongTextArea</type>
        <visibleLines>30</visibleLines>
    </fields>
    <label>MyFirstObject</label>
    <nameField>
        <label>MyFirstObject Name</label>
        <type>Text</type>
    </nameField>
    <pluralLabel>MyFirstObjects</pluralLabel>
    <sharingModel>ReadWrite</sharingModel>
</CustomObject>
```

The following is the metadata definition of an external object for Salesforce Connect.

```
<?xml version="1.0" encoding="UTF-8"?>
<CustomObject xmlns="http://soap.sforce.com/2006/04/metadata">
    <actionOverrides>
        <actionName>CancelEdit</actionName>
        <type>Default</type>
    </actionOverrides>
    <actionOverrides>
        <actionName>Delete</actionName>
        <type>Default</type>
```

```
    </actionOverrides>
    <actionOverrides>
        <actionName>Edit</actionName>
        <type>Default</type>
    </actionOverrides>
    <actionOverrides>
        <actionName>Follow</actionName>
        <type>Default</type>
    </actionOverrides>
    <actionOverrides>
        <actionName>List</actionName>
        <type>Default</type>
    </actionOverrides>
    <actionOverrides>
        <actionName>New</actionName>
        <type>Default</type>
    </actionOverrides>
    <actionOverrides>
        <actionName>SaveEdit</actionName>
        <type>Default</type>
    </actionOverrides>
    <actionOverrides>
        <actionName>Tab</actionName>
        <type>Default</type>
    </actionOverrides>
    <actionOverrides>
        <actionName>View</actionName>
        <type>Default</type>
    </actionOverrides>
    <deploymentStatus>InDevelopment</deploymentStatus>
    <description>Products</description>
    <enableFeeds>false</enableFeeds>
    <externalDataSource>OData</externalDataSource>
    <externalIndexAvailable>false</externalIndexAvailable>
    <externalName>Products</externalName>
    <fields>
        <fullName>DiscontinuedDate__c</fullName>
        <description>DiscontinuedDate</description>
        <externalDeveloperName>DiscontinuedDate</externalDeveloperName>
        <externalId>false</externalId>
        <isFilteringDisabled>false</isFilteringDisabled>
        <isNameField>false</isNameField>
        <isSortingDisabled>false</isSortingDisabled>
        <label>DiscontinuedDate</label>
        <required>false</required>
        <type>DateTime</type>
    </fields>
    <fields>
        <fullName>ID__c</fullName>
        <description>ID</description>
        <externalDeveloperName>ID</externalDeveloperName>
        <externalId>false</externalId>
        <isFilteringDisabled>false</isFilteringDisabled>
        <isNameField>false</isNameField>
```

```
    <isSortingDisabled>false</isSortingDisabled>
    <label>ID</label>
    <precision>18</precision>
    <required>false</required>
    <scale>0</scale>
    <type>Number</type>
    <unique>false</unique>
</fields>
<fields>
    <fullName>Name__c</fullName>
    <description>Name</description>
    <externalDeveloperName>Name</externalDeveloperName>
    <externalId>false</externalId>
    <isFilteringDisabled>false</isFilteringDisabled>
    <isNameField>false</isNameField>
    <isSortingDisabled>false</isSortingDisabled>
    <label>Name</label>
    <length>128</length>
    <required>false</required>
    <type>Text</type>
    <unique>false</unique>
</fields>
<fields>
    <fullName>Price__c</fullName>
    <description>Price</description>
    <externalDeveloperName>Price</externalDeveloperName>
    <externalId>false</externalId>
    <isFilteringDisabled>false</isFilteringDisabled>
    <isNameField>false</isNameField>
    <isSortingDisabled>false</isSortingDisabled>
    <label>Price</label>
    <precision>16</precision>
    <required>false</required>
    <scale>2</scale>
    <type>Number</type>
    <unique>false</unique>
</fields>
<fields>
    <fullName>Products__c</fullName>
    <externalDeveloperName>Products</externalDeveloperName>
    <externalId>false</externalId>
    <isFilteringDisabled>false</isFilteringDisabled>
    <isNameField>false</isNameField>
    <isSortingDisabled>false</isSortingDisabled>
    <label>Products</label>
    <length>20</length>
    <referenceTo>Products__x</referenceTo>
    <relationshipLabel>Products</relationshipLabel>
    <relationshipName>Products</relationshipName>
    <type>ExternalLookup</type>
</fields>
<fields>
    <fullName>Rating__c</fullName>
    <description>Rating</description>
```

```
        <externalDeveloperName>Rating</externalDeveloperName>
        <externalId>false</externalId>
        <isFilteringDisabled>false</isFilteringDisabled>
        <isNameField>false</isNameField>
        <isSortingDisabled>false</isSortingDisabled>
        <label>Rating</label>
        <precision>18</precision>
        <required>false</required>
        <scale>0</scale>
        <type>Number</type>
        <unique>false</unique>
    </fields>
    <fields>
        <fullName>ReleaseDate__c</fullName>
        <description>ReleaseDate</description>
        <externalDeveloperName>ReleaseDate</externalDeveloperName>
        <externalId>false</externalId>
        <isFilteringDisabled>false</isFilteringDisabled>
        <isNameField>false</isNameField>
        <isSortingDisabled>false</isSortingDisabled>
        <label>ReleaseDate</label>
        <required>false</required>
        <type>DateTime</type>
    </fields>
    <label>Products</label>
    <pluralLabel>Products</pluralLabel>
    <searchLayouts>
        <customTabListAdditionalFields>ExternalId</customTabListAdditionalFields>
        <lookupDialogsAdditionalFields>ExternalId</lookupDialogsAdditionalFields>
       <lookupPhoneDialogsAdditionalFields>ExternalId</lookupPhoneDialogsAdditionalFields>

        <searchResultsAdditionalFields>ExternalId</searchResultsAdditionalFields>
        <searchResultsAdditionalFields>DisplayUrl</searchResultsAdditionalFields>
        <searchResultsAdditionalFields>ID__c</searchResultsAdditionalFields>
    </searchLayouts>
</CustomObject>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file for Field Sets and Record Types but not for other components. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

1. ActionOverride

   Represents an action override on a standard or custom object. Use it to create, update, edit, or delete action overrides. You can access ActionOverride only by accessing its encompassing CustomObject.

2. BusinessProcess

   The BusinessProcess metadata type enables you to display different picklist values for users based on their profile.

3. CompactLayout

   Represents the metadata associated with a compact layout. This type extends the Metadata metadata type and inherits its `fullName` field.

4. CustomField

   Represents the metadata associated with a field. Use this metadata type to create, update, or delete custom field definitions on standard, custom, and external objects or standard field definitions on standard objects.

5. FieldSet

   Represents a field set. A field set is a grouping of fields. For example, you could have a field set that contains fields describing a user's first name, middle name, last name, and business title.

6. HistoryRetentionPolicy

   Represents the policy for archiving field history data. When you set a policy, you specify the number of months that you want to keep field history in Salesforce before archiving it. By default, when Field Audit Trail is enabled, all field history is retained.

7. Index

   Represents an index defined within a custom big object. Use this metadata type to define the composite primary key (index) for a custom big object.This type extends the Metadata metadata type and inherits its `fullName` field.

8. ListView

   ListView allows you to see a filtered list of records, such as contacts, accounts, or custom objects.

9. NamedFilter

   Represents the metadata associated with a lookup filter. This metadata type is used to create, update, or delete lookup filter definitions. This component has been removed as of API version 30.0 and is only available in previous API versions. The metadata associated with a lookup filter is now represented by the lookupFilter field in the CustomField component.

10. Picklist (Including Dependent Picklist)

    Deprecated. Represents a picklist (or dependent picklist) definition for a custom field in a custom object or a custom or standard field in a standard object, such as an account.

11. ProfileSearchLayouts

    Represents a user profile's search results layouts for an object. `ProfileSearchLayouts` are similar to `SearchLayouts`. However, with profile-specific layouts, each user profile can have a different search results layout for an object.

12. RecordType

    Represents the metadata associated with a record type. Record types let you offer different business processes, picklist values, and page layouts to different users. Use this metadata type to create, update, or delete record type definitions for a custom object.

13. SearchLayouts

    Represents the metadata associated with the search layouts for an object. You can customize which fields to display for users in search results, search filter fields, lookup dialogs, and recent record lists on tab home pages. You can access SearchLayouts only by accessing its encompassing CustomObject.

14. SharingReason

    Represents an Apex sharing reason, which is used to indicate why sharing was implemented for a custom object. Apex managed sharing allows developers to use Apex to programmatically share custom objects. When you use Apex managed sharing to share a custom object, only users with the "Modify All Data" permission can add or change the sharing on the custom object's record, and the sharing access is maintained across record owner changes.

15. SharingRecalculation

    Represents Apex classes that recalculate the Apex managed sharing for a specific custom object.

16. ValidationRule

    Represents a validation rule, which is used to verify that the data a user enters in a record is valid and can be saved. A validation rule contains a formula or expression that evaluates the data in one or more fields and returns a value of `true` or `false`. Validation rules also include an error message that your client application can display to the user when the rule returns a value of `true` due to invalid data.

17. WebLink
    Represents a custom button or link defined in a custom object.
18. Metadata Field Types
    These field types extend the field types described in the *Salesforce Object Reference*.

SEE ALSO:

- CustomField
- Metadata
- Picklist (Including Dependent Picklist)
- SearchLayouts
- WebLink
- CustomObjectTranslation
- ListView
- CompactLayout

ActionOverride

Represents an action override on a standard or custom object. Use it to create, update, edit, or delete action overrides. You can access ActionOverride only by accessing its encompassing CustomObject.

Declarative Metadata File Suffix and Directory Location

Action overrides are defined as part of a standard or custom object.

Version

Action overrides are available in API version 18.0 and later. As of Summer '13, action overrides can be applied to both standard and custom objects. Previously, action overrides only applied to custom objects.

Fields

Unless otherwise noted, all fields are creatable, filterable, and nillable.

Field Name	Field Type	Description
`actionName`	string	Required. The possible values are the same as the actions you can override: • `accept` • `clone` • `delete` • `edit` • `list` • `new` • `tab` • `view`

Field Name	Field Type	Description
`comment`	string	Any comments you want associated with the override.
`content`	string	Set this field if `type` is set to `flexipage`, `lightningcomponent`, `scontrol`, or `visualforce`. It refers to the name of the Lightning page, Lightning component, s-control, or Visualforce page to use as the override. To reference installed components, use this format: ***`Component_namespace__Component_name`***.
`formFactor`	FormFactor (enumeration of type string)	The size of the page being overridden. If the `type` field is set to `flexipage`, set this field to `Large` to override the View action with a Lightning page in Lightning Experience. The `Large` value represents the Lightning Experience desktop environment and is valid only for the `flexipage` and `lightningcomponent` types. The `Small` value represents the Salesforce mobile app on a phone or tablet. The `Medium` value is reserved for future use. The `null` value (which is the same as specifying no value) represents Salesforce Classic. This field is available in API version 37.0 and later and is part of the feature for creating and editing record pages in Lightning Experience. Lightning component overrides return different `FormFactor` values depending on the API version used. • In API version 41.0 and earlier, Lightning component overrides return only the `null` value (no value), representing the Salesforce Classic environment. • In API version 42.0, if you specify different Lightning component overrides for Lightning Experience and mobile, one component is selected randomly for both overrides and its `FormFactor` value is returned. If there's a conflict between Lightning components, and a Visualforce page override is also specified for Salesforce Classic, the Visualforce page takes precedence. • In API version 43.0 and later, a Lightning component override for Lightning Experience returns the `Large` value and a Lightning component override for mobile returns the `Small` value, as expected.
`skipRecordTypeSelect`	boolean	Set this field to `true` if you prefer that any new records created by this action override aren't forwarded to the record type selection page. This field is only valid if the `actionName` is a "create" type (like `new`), and `type` is set to `visualforce`. This field is available in API version 21.0 and later.
`type`	ActionOverrideType (enumeration of type string)	Required. Represents the type of action override. Valid values are described in ActionOverrideType.

ActionOverrideType

ActionOverrideType on page 680 is an enumeration of type string that defines which kind of action override to use. The valid values are:

- `default`—The override uses a custom override provided by an installed package. If there isn't one available, the standard Salesforce behavior is used.
- `flexipage`—The override uses behavior from a Lightning page, and is only valid for the View action in Lightning Experience.
- `lightningcomponent`—The override uses behavior from a Lightning component.
- `scontrol`—The override uses behavior from an s-control.
- `standard`—The override uses regular Salesforce behavior.
- `visualforce`—The override uses behavior from a Visualforce page.

Note: Existing s-controls can be used as overrides for Salesforce Classic under certain conditions. However, s-controls have been deprecated since the Spring '09 release. We recommend using Visualforce pages instead.

Usage

You can't delete ActionOverrides by deploying with `destructiveChange.xml`. To delete an ActionOverride, retrieve the CustomObject. In the definition file, find the `<ActionOverrides>` section, and remove the `<content>` row. Then, change the `<type>` value in that same section to `Default`. Do this for every override you want to reset. After making the changes, rezip the folder and deploy.

You can remove one override at a time each with its own deploy, or you can remove multiple overrides in a single deploy. However, we recommend that you do a fresh retrieve every time you want to delete a new override. Don't use a previously retrieved file.

Org default flexipage override assignment metadata can't be retrieved from a managed package.

Declarative Metadata Sample Definitions

You can define action overrides, as in these examples for the Edit action.

A Visualforce page override for Salesforce Classic:

```
<CustomObject xmlns="http://soap.sforce.com/2006/04/metadata">
    <actionOverrides>
        <actionName>edit</actionName>
        <type>visualforce</type>
        <content>myEditVFPage</content>
        <comment>This edit action is a lot safer.</comment>
     </actionOverrides>
</CustomObject
```

This example includes no value for FormFactor. Using no value is the same as using the `null` value, which represents Salesforce Classic.

A Lightning component override for Lightning Experience:

```
<CustomObject xmlns="http://soap.sforce.com/2006/04/metadata">
    <actionOverrides>
        <actionName>edit</actionName>
        <type>lightningcomponent</type>
        <content>myEditLightningComponent</content>
        <formFactor>Large</formFactor>
        <comment>This edit action is a lot safer.</comment>
```

```
     </actionOverrides>
</CustomObject>
```

A Lightning component override for the Salesforce mobile app:

```
<CustomObject xmlns="http://soap.sforce.com/2006/04/metadata">
    <actionOverrides>
        <actionName>edit</actionName>
        <type>lightningcomponent</type>
        <content>myEditLightningComponent</content>
        <formFactor>Small</formFactor>
        <comment>This edit action is a lot safer.</comment>
     </actionOverrides>
</CustomObject>
```

When overrides are included in a managed package, the overrides are represented as `default` type in the metadata. Calling retrieve() presents the following:

```
<CustomObject xmlns="http://soap.sforce.com/2006/04/metadata">
    <actionOverrides>
        <actionName>edit</actionName>
        <type>default</type>
     </actionOverrides>
</CustomObject>
```

If you subscribe to a managed package with default overrides, you can replace the default override behavior by editing the XML. For example, to replace the Visualforce page override with the Salesforce standard page for Salesforce Classic, use:

```
<CustomObject xmlns="http://soap.sforce.com/2006/04/metadata">
    <actionOverrides>
        <actionName>edit</actionName>
        <type>standard</type>
     </actionOverrides>
</CustomObject>
```

To set a Lightning page action override on the View standard button in Lightning Experience, use:

```
<CustomObject xmlns="http://soap.sforce.com/2006/04/metadata">
    <actionOverrides>
        <actionName>View</actionName>
        <content>myLightningPage</content>
        <formFactor>Large</formFactor>
        <type>flexipage</type>
    </actionOverrides>
</CustomObject>
```

Wildcard Support in the Manifest File

This metadata type doesn't support the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SEE ALSO:

CustomObject

BusinessProcess

The BusinessProcess metadata type enables you to display different picklist values for users based on their profile.

Multiple business processes allow you to track separate sales, support, and lead lifecycles. A sales, support, lead, or solution process is assigned to a record type. The record type determines the user profiles that are associated with the business process.

Important: Don't use business processes as an access control mechanism. Profile assignment governs create and edit access for business process but doesn't govern read access. For example, a user assigned to a profile that isn't enabled for a particular business process can't create or edit it, but they can read the business process record.

Users with access to a business process can read all information it stores. Don't store sensitive information in the business process description, name, or picklist values. Instead, store sensitive information in a separate object or fields to which you've applied appropriate access controls.

Declarative Metadata File Suffix and Directory Location

Business processes are defined as part of the custom object or standard object definition. See CustomObject for more information.

Version

BusinessProcess on page 682 components are available in API version 17.0 and later.

Special Access Rules

Access to this object requires the View Setup and Configuration permission.

Fields

Field	Field Type	Description
`description`	string	Description for the business process.
`fullName`	string	Required. The name used as a unique identifier for API access. This field is inherited from the Metadata component, but the string it contains is created differently than the `fullName` strings for other types. For a `fullName` string BusinessProcess on page 682, the `fullName` is created combining the Entity Name and Business Process Name. For example, for a business process called "Bulk Orders" for opportunities, the `fullName` would be `Opportunity.Bulk Orders`.
`isActive`	boolean	Indicates if the business process is active (`true`) or not (`false`).
`namespacePrefix`	string	The namespace of the developer organization where the package was created.
`values`	PicklistValue[]	A list of picklist values associated with this business process.

Declarative Metadata Sample Definition

The following is a sample XML definition of a lead business process.

```
<?xml version="1.0" encoding="UTF-8"?>
<CustomObject xmlns="http://soap.sforce.com/2006/04/metadata">
....
    <businessProcesses>
        <fullName>HardwareLeadProcess</fullName>
        <description>Lead Process for hardware division</description>
        <isActive>true</isActive>
        <values>
            <fullName>Closed - Converted</fullName>
            <default>false</default>
        </values>
        <values>
            <fullName>CustomLeadStep1</fullName>
            <default>false</default>
        </values>
        <values>
            <fullName>CustomLeadStep2</fullName>
            <default>false</default>
        </values>
        <values>
            <fullName>Open - Not Contacted</fullName>
            <default>false</default>
        </values>
        <values>
            <fullName>Working - Contacted</fullName>
            <default>true</default>
        </values>
    </businessProcesses>
....
</CustomObject>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file only when a RecordType on page 717 is specified. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SEE ALSO:

Salesforce DX Developer Guide: BusinessProcessGroup

CustomObject

CompactLayout

Represents the metadata associated with a compact layout. This type extends the Metadata metadata type and inherits its `fullName` field.

A compact layout displays a record's key fields at a glance in the Salesforce mobile app, Lightning Experience, and in the Outlook and Gmail integrations.

Compact layouts support all field types except:

- text area
- long text area
- rich text area
- multi-select picklist

For more information on compact layouts, see Compact Layouts in the Salesforce Help.

File Suffix and Directory Location

Compact layouts are defined as part of the custom object, standard object, or external object definition. See CustomObject for more information.

Version

CompactLayout components are available in API version 29.0 and later. CompactLayout components are available for external objects in API version 42.0 and later.

Fields

Field Name	Field Type	Description
`fields`	string	The fields assigned to the compact layout. Their order represents the prioritization given to them when defining the compact layout.
`label`	string	Label that represents the object throughout the Salesforce user interface.

Declarative Metadata Sample Definition

The following is an example of a CompactLayout component:

```
<CustomObject xmlns="http://soap.sforce.com/2006/04/metadata">
    <actionOverrides>
        <actionName>Accept</actionName>
        <type>Default</type>
    </actionOverrides>
    <actionOverrides>
        <actionName>Clone</actionName>
        <type>Default</type>
    </actionOverrides>
    <actionOverrides>
        <actionName>Delete</actionName>
        <type>Default</type>
    </actionOverrides>
    <actionOverrides>
        <actionName>Edit</actionName>
        <type>Default</type>
    </actionOverrides>
    <actionOverrides>
        <actionName>List</actionName>
        <type>Default</type>
    </actionOverrides>
```

```
    <actionOverrides>
        <actionName>New</actionName>
        <type>Default</type>
    </actionOverrides>
    <actionOverrides>
        <actionName>Tab</actionName>
        <type>Default</type>
    </actionOverrides>
    <actionOverrides>
        <actionName>View</actionName>
        <type>Default</type>
    </actionOverrides>
    <compactLayouts>
        <fullName>testCompactLayout</fullName>
        <fields>textfield__c</fields>
        <label>testCompactLayoutLabel</label>
    </compactLayouts>
    <compactLayoutAssignment>SYSTEM</compactLayoutAssignment>
    <deploymentStatus>Deployed</deploymentStatus>
    <enableActivities>false</enableActivities>
    <enableFeeds>false</enableFeeds>
    <enableHistory>false</enableHistory>
    <enableReports>false</enableReports>
    <fields>
        <fullName>textfield__c</fullName>
        <externalId>false</externalId>
        <label>textfield</label>
        <length>255</length>
        <required>false</required>
        <type>Text</type>
        <unique>false</unique>
    </fields>
    <label>customObj</label>
    <nameField>
        <label>customObj Name</label>
        <type>Text</type>
    </nameField>
    <pluralLabel>customObjs</pluralLabel>
    <recordTypes>
        <fullName>RT1</fullName>
        <active>true</active>
        <label>RT1</label>
        <compactLayoutAssignment>testCompactLayout</compactLayoutAssignment>
    </recordTypes>
    <recordTypes>
        <fullName>RT2</fullName>
        <active>true</active>
        <label>RT2</label>
    </recordTypes>
    <searchLayouts/>
    <sharingModel>ReadWrite</sharingModel>
</CustomObject>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

CustomField

Represents the metadata associated with a field. Use this metadata type to create, update, or delete custom field definitions on standard, custom, and external objects or standard field definitions on standard objects.

This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Only standard fields that you can customize are supported, that is, standard fields to which you can add help text or enable history tracking or Chatter feed tracking. Other standard fields aren't supported, including system fields (such as `CreatedById` or `LastModifiedDate`) and autonumber fields. Some standard picklist fields aren't supported. See Unsupported Metadata Types. By default, a custom object doesn't have any standard fields that are customizable.

Specify the full name whenever you create or update a field. For example, a custom field on a custom object:

```
MyCustomObject__c.MyCustomField__c
```

An example of a custom field on a standard object:

```
Account.MyAcctCustomField__c
```

An example of a standard field on a standard object:

```
Account.Phone
```

An example of a custom field on an external object:

```
MyExternalObject__x.MyCustomField__c
```

Note: In Metadata API, external objects are represented by the CustomObject metadata type.

These custom field types aren't available for external objects.

- Auto-number (available only with the cross-org adapter for Salesforce Connect)
- Currency (available only with the cross-org adapter for Salesforce Connect)
- Formula
- Location
- Master-detail relationship
- Picklist and multi-select picklist (available only with the cross-org adapter for Salesforce Connect)
- Roll-up summary
- Text (encrypted)
- Text Area (rich)

Declarative Metadata File Suffix and Directory Location

Custom fields are user-defined fields and are part of the custom object or standard object definition. See CustomObject for more information. Standard fields are predefined on standard objects.

Note: Retrieving a component of this metadata type in a project makes the component appear in any Profile and PermissionSet components that are retrieved in the same package.

Retrieving Fields on Custom or Standard Objects

When you retrieve a custom or standard object, you return everything associated with the object, except for standard fields that aren't customizable. You can also retrieve only specific fields for an object by explicitly naming the object and fields in `package.xml`. The following definition in `package.xml` creates the files `objects/MyCustomObject__c.object` and `objects/Account.object`, each containing the requested field definitions.

```
<types>
  <members>MyCustomObject__c.MyCustomField__c</members>
  <members>Account.MyCustomAccountField__c</members>
  <members>Account.Phone</members>
  <name>CustomField</name>
</types>
```

Retrieving or Deploying Fields on Data Cloud Objects

When you retrieve a Data Cloud object, such as a DLO or DMO, not all of the custom field properties are returned. The properties returned depend on the data type of the custom field.

When you deploy a Data Cloud object via Metadata API, in API version 60.0 or later, the call succeeds only if the properties are supported by the custom field's data type. If you include a property that isn't supported by the field's data type, the API returns an error.

Data Cloud objects support these data types.

- Boolean/Checkbox
- Date
- DateTime
- Email
- Lookup (DMOs only)
- Number
- Percent
- Phone
- Text
- Url

Version

Custom and standard fields are available in API version 10.0 and later.

Fields

Unless otherwise noted, all fields are creatable, filterable, and nillable.

Field Name	Field Type	Description
`businessOwnerGroup`	reference	Indicates the group associated with this field. The business owner group understands the importance of the field's data to your company, and can be responsible for determining the minimum security classification. This field is available in API version 45.0 and later.
`businessOwnerUser`	reference	Indicates the person associated with this field. The business owner understands the importance of the field's data to your company, and can be responsible for determining the minimum security classification. This field is available in API version 45.0 and later.
`businessStatus`	picklist	Indicates whether the field is in use. Valid values include: • `Active` • `DeprecateCandidate` • `Hidden` This field is available in API version 45.0 and later
`caseSensitive`	boolean	Indicates whether the field is case-sensitive (`true`) or not (`false`). For indirect lookup relationship fields on external objects, this attribute affects how this custom field's values are matched against the values of the `referenceTargetField`.
`complianceGroup`	multipicklist	Indicates the compliance acts, definitions, or regulations related to the field's data. Valid values include: • `CCPA` • `COPPA` • `GDPR` • `HIPAA` • `PCI` • `PII` This field is available in API version 47.0 and later.
`customDataType`	string	Deprecated in the Spring '19 (API version 45.0) release.
`defaultValue`	string	If specified, represents the default value of the field.
`deleteConstraint`	DeleteConstraint (enumeration of type string)	Provides deletion options for lookup relationships. Valid values are: • `Cascade`—Deletes the lookup record as well as associated lookup fields. • `Restrict`—Prevents the record from being deleted if it's in a lookup relationship. • `SetNull`—This value is the default. If the lookup record is deleted, the lookup field is cleared.

Field Name	Field Type	Description
		For more information on lookup relationships, see "Object Relationships" in Salesforce Help.
`deprecated`	boolean	Reserved for future use.
`description`	string	Description of the field.
`displayFormat`	string	The display format.
`displayLocationInDecimal`	boolean	Indicates how the geolocation values of a custom Location field appear in the user interface. If `true`, the geolocation values appear in decimal notation. If `false`, the geolocation values appear as degrees, minutes, and seconds.
`elementType`	ElementType (enumeration of type string)	Reserved for future use.
`encrypted`	boolean	This entry is about Shield Platform Encryption, not Classic Encryption. Indicates whether this field is encrypted (`true`) or not (`false`). This field is available in API version 34.0 through 43.0.
`encryptionScheme`	EncryptionScheme (enumeration of type string)	This entry is about Shield Platform Encryption, not Classic Encryption. For encrypted fields, determines which encryption scheme a field takes. Valid values are • `CaseInsensitiveDeterministicEncryption` • `CaseSensitiveDeterministicEncryption` • `None` • `ProbabilisticEncryption` This field is available in API version 44.0 and later.
`externalDeveloperName`	string	Available only for external objects. Name of the table column on the external data source that maps to this custom field in Salesforce. Corresponds to `External Column Name` in the user interface. This field is available in API version 32.0 and later.
`externalId`	boolean	Indicates whether the field is an external ID field (`true`) or not (`false`). This property is returned only if the custom field data type is AutoNumber, Email, Number, or Text.
`fieldManageability`	FieldManageability (enumeration of type string)	Determines who can update the field after it's released in a managed package. Valid values: • `Locked`—The field can't be updated. • `DeveloperControlled`—The creator of the record can update the field with a package upgrade.

Field Name	Field Type	Description
		• `SubscriberControlled`—Anyone with proper permissions can update the field. The field can't be updated with a package upgrade. Available only for fields on custom metadata types. If the field type is `MetadataRelationship`, and the manageability of the entity definition field is: • Subscriber-controlled, then the Field Definition field must be subscriber-controlled. • Upgradeable, then the Field Definition field must be either upgradeable or subscriber-controlled.
`formula`	string	If specified, represents a formula on the field.
`formulaTreatBlanksAs`	TreatBlanksAs (enumeration of type string)	Indicates how to treat blanks in a formula. Valid values are: `BlankAsBlank` and `BlankAsZero`.
`fullName`	string	Inherited from Metadata, this field is defined in the WSDL for this metadata type. It must be specified when creating, updating, or deleting. See `createMetadata()` to see an example of this field specified for a call. This value can't be `null`.
`globalPicklist`	string.	(This field is available in API version 37.0 only and removed from later versions.) If this custom field is a picklist that's based on a global picklist, `globalPicklist` is the name of the global picklist whose value set this picklist inherits. A custom picklist that's based on a global picklist is restricted. You can only add or remove values by editing the global picklist.
`indexed`	boolean	Indicates if the field is indexed. If this field is unique or the `externalId` is set true, the `isIndexed` value is set to true. This field has been deprecated as of API version 14.0 and is only provided for backward compatibility.
`inlineHelpText`	string	Represents the content of field-level help. For more information, see "Define Field-Level Help" in Salesforce Help.
`isAIPredictionField`	boolean	Available for Number type custom fields when you use Einstein Prediction Builder. Denotes whether the field can store and display Einstein prediction data on an object. Use Einstein Prediction Builder to determine the data for the target field. This field is available in API version 43.0 and later.
`isFilteringDisabled`	boolean	Available only for external objects. Indicates whether the custom field is available in filters. This field is available in API version 32.0 and later.
`isNameField`	boolean	Available only for external object fields of type text. For each external object, you can specify one field as the name field. If you

Field Name	Field Type	Description
		set this value to `true`, make sure that the external table column identified by the `externalDeveloperName` attribute contains name values. This field is available in API version 32.0 and later.
`isSortingDisabled`	boolean	Available only for external objects. Indicates whether the custom field is sortable. This field is available in API version 32.0 and later.
`label`	string	Label for the field. You can't update the label for standard picklist fields, such as the `Industry` field for accounts.
`length`	int	Length of the field.
`lookupFilter`	LookupFilter	Represents the metadata associated with a lookup filter. This metadata type is used to create, update, or delete lookup filter definitions. This component has been removed as of API version 30.0 and is only available in previous API versions. The metadata associated with a lookup filter is now represented by the `lookupFilter` field in the CustomField component. This field is available in API version 30.0 and later. LookupFilter isn't supported on the article type object.
`maskChar`	EncryptedFieldMaskChar (enumeration of type string)	This page is about Classic Encryption, not Shield Platform Encryption. For encrypted fields, specifies the character to be used as a mask. Valid values are: • `asterisk` • `X` For more information on encrypted fields, see Classic Encryption for Custom Fields in Salesforce Help.
`maskType`	EncryptedFieldMaskType (enumeration of type string)	This page is about Classic Encryption, not Shield Platform Encryption. For encrypted text fields, specifies the format of the masked and unmasked characters in the field. Valid values are: • `all`—All characters in the field are hidden. This option is equivalent to the `Mask All Characters` option in Salesforce. • `creditCard`—The first 12 characters are hidden and the last four display. This option is equivalent to the `Credit Card Number` option in Salesforce. • `lastFour`—All characters are hidden but the last four display. This option is equivalent to the `Last Four Characters Clear` option in Salesforce.

Field Name	Field Type	Description
		• `nino`—All characters are hidden. Salesforce automatically inserts spaces after each pair of characters if the field contains nine characters. This option is equivalent to the `National Insurance Number` option in Salesforce. • `sin`—All characters are hidden but the last four display. This option is equivalent to the `Social Insurance Number` option in Salesforce. • `ssn`—The first five characters are hidden and the last four display. This option is equivalent to the `Social Security Number` option in Salesforce. For more information on encrypted fields, see "Classic Encryption for Custom Fields" in Salesforce Help.
`metadataRelationshipControllingField`	string	In custom metadata relationships, represents the controlling field that specifies the standard or custom object in an entity definition metadata relationship. Required when creating a field definition or entity particle metadata relationship on a custom metadata type. The object specified in the controlling field determines the values available in its dependent field definition or entity particle. For example, specifying the Account object filters the available fields in the field definition to Account fields only. This field is available in API version 39.0 and later.
`picklist`	Picklist	(**Deprecated.** Use this field in API version 37.0 and earlier only. In later versions, use `valueSet` instead.) If specified, the field is a picklist, and this field enumerates the picklist values and labels.
`populateExistingRows`	boolean	Indicates whether existing rows are going to be populated (`true`) or not (`false`).
`precision`	int	The precision for number values. Precision is the number of digits in a number. For example, the number 256.99 has a precision value of 5.
`referenceTargetField`	string	Available only for indirect lookup relationship fields on external objects. Specifies the custom field on the parent object to match against this indirect lookup relationship field, whose values come from an external data source. The specified custom field on the parent object must have both `externalId` and `unique` set to `true`. This field is available in API version 32.0 and later.
`referenceTo`	string	If specified, indicates a reference this field has to another object.
`relationshipLabel`	string	Label for the relationship.
`relationshipName`	string	If specified, indicates the value for one-to-many relationships. For example, in the object MyObject that had a relationship to YourObject, the relationship name can be YourObjects.

Field Name	Field Type	Description
`relationshipOrder`	int	This field is valid for all master-detail relationships, but the value is only non-zero for junction objects. A junction object has two master-detail relationships, and is analogous to an association table in a many-to-many relationship. Junction objects must define one parent object as primary (0), the other as secondary (1). The definition of primary or secondary affects delete behavior and inheritance of look and feel, and record ownership for junction objects. For more information, see Salesforce Help. 0 or 1 are the only valid values, and 0 is always the value for objects that aren't junction objects.
`reparentableMasterDetail`	boolean	Indicates whether the child records in a master-detail relationship on a custom object can be reparented to different parent records. The default value is `false`. This field is available in API version 25.0 and later.
`required`	boolean	Indicates whether the field requires a value on creation (`true`) or not (`false`).
`scale`	int	The scale for the field. Scale is the number of digits to the right of the decimal point in a number. For example, the number 256.99 has a scale of 2.
`securityClassification`	picklist	Indicates the sensitivity of the data contained in the field. Valid values include: • `Public` • `Internal` • `Confidential` • `Restricted` • `MissionCritical` This field is available in API version 45.0 and later.
`startingNumber`	int	If specified, indicates the starting number for the field. When you create records, `Starting Number`'s value increments to store the number that will be assigned to the next auto-number field created. • You can't retrieve the starting number of an auto-number field through Metadata API. To specify a `Starting Number` while deploying, add a `startingNumber` tag for your field to your `package.xml` file. For example: `<startingNumber>42</startingNumber>` • If you deploy without specifying a `Starting Number` value in your `package.xml` file, the default starting number for standard fields is `0`. The default starting number for custom fields is `1`.

Field Name	Field Type	Description
`stripMarkup`	boolean	Set to `true` to remove markup, or `false` to preserve markup. Used when converting a rich text area to a long text area.
`summarizedField`	string	Represents the field on the detail row that's being summarized. This field can't be null unless the `summaryOperation` value is `count`.
`summaryFilterItems`	FilterItem[]	Represents the set of filter conditions for this field if it's a summary field. This field is summed on the child if the filter conditions are met.
`summaryForeignKey`	string	Represents the master-detail field on the child that defines the relationship between the parent and the child.
`summaryOperation`	SummaryOperations (enumeration of type string)	Represents the type of sum operation to be performed. Valid values are: • `Count` • `Min` • `Max` • `Sum`
`trackFeedHistory`	boolean	Indicates whether the field is enabled for feed tracking (`true`) or not (`false`). To set this field to `true`, the `enableFeeds` field on the associated CustomObject must also be `true`. For more information, see "Customize Chatter Feed Tracking" in Salesforce Help. This field is available in API version 18.0 and later.
`trackHistory`	boolean	Indicates whether history tracking is enabled for the field (`true`) or not (`false`). Also available for standard object fields (picklist and lookup fields only) in API version 30.0 and later. To set `trackHistory` to `true`, the `enableHistory` field on the associated standard or custom object must also be `true`. For more information, see "Field History Tracking" in Salesforce Help. Field history tracking isn't available for external objects.
`trackTrending`	boolean	Indicates whether historical trending data is captured for the field (`true`) or not (`false`). An object is enabled for historical trending if this attribute is `true` for at least one field. Available in API version 29.0 and later. For more information, see "Report on Historical Changes" in Salesforce Help.

Field Name	Field Type	Description
`trueValueIndexed`	boolean	Only relevant for a checkbox field. If set, `true` values are built into the index. This field has been deprecated as of API version 14.0 and is only provided for backward compatibility.
`type`	FieldType (enumeration of type string)	Indicates the field type for the field. Valid values are enumerated in FieldType. For standard fields on standard objects, the `type` field is optional. This field is included for some standard field types, such as Picklist or Lookup, but not for others. The `type` field is included for custom fields.
`unique`	boolean	Indicates whether the field is unique (`true`) or not (`false`).
`valueSet`	ValueSet	Represents the set of values that make up a picklist on a custom field. Each value is defined as a CustomValue on page 761. If this custom field is a picklist that uses a global value set, `valueSet` is the name of the global value set whose values this picklist inherits. A custom picklist that uses a global value set is restricted. You can only add or remove values by editing the global value set. A ValueSet component has either a `valueSetDefinition` or a `valueName` specified, but never both. This field is available in API version 38.0 and later.
`visibleLines`	int	Indicates the number of lines displayed for the field.
`writeRequiresMasterRead`	boolean	Sets the minimum sharing access level required on the primary record to create, edit, or delete child records. This field applies only to master-detail or junction object custom field types. • `true`—Allows users with Read access to the primary record permission to create, edit, or delete child records. This setting makes sharing less restrictive. • `false`—Allows users with Read/Write access to the primary record permission to create, edit, or delete child records. This setting is more restrictive than `true`, and is the default value. For junction objects, the most restrictive access from the two parents is enforced. For example, if you set to `true` on both master-detail fields, but users have Read access to one primary record and Read/Write access to the other primary record, users aren't able to create, edit, or delete child records.

Fields use additional data types. For more information, see Metadata Field Types on page 730.

MktDataModelFieldAttributes

This is a subtype of CustomField.

Field Name	Field Type	Description
definitionCreationType	DefinitionCreationType enumeration	Indicates how this object was added. Valid values are: • Bridge • Custom • Derived • Standard • System Valid values availble in API version 62.0 and later are: • Activation_Audience • Ad_Audience_Insights • ADG • Calculated_Insight • CG_Audience • Chunk • Directory_Table • External • Problem_Records • Segment_Membership • Semantic • Transform • Vector_Embedding
invalidMergeActionType	InvalidMergeActionType (enumeration of type string)	If this field is used for merging data, indicates what the system should do when an invalid merge occurs. Valid values are: • Drop • Keep • Override
isDynamicLookup	boolean	When true, the existing data is queried for a unique set of values for this field.
primaryIndexOrder	int	If supplied, indicates that this field is part of the primary key. The number value (starting at 1) indicates the order of attributes if it's a compound primary key.
refAttrDeveloperName	string	When this is a Standard Field, it's the Name of the field from the Reference Model.
mktDatalakeSrcKeyQualifier	string	String storing the developer name of MktDataLakeSrcKeyQualifier configured on the field

MktDataLakeFieldAttributes

This is a subtype of CustomField. MktDataLakeFieldAttributes is available in API version 50.0 or later.

Field Name	Field Type	Description
definitionCreationType	DefinitionCreationType (enumeration of type string)	Indicates how this object is added. Valid values are: • Bridge • Custom • Derived • Standard • System Valid values available in API version 62.0 and later are: • ADG • Calculated_Insight • CG_Audience • Chunk • Directory_Table • External • Semantic • Vector_Embedding
dateFormat	string	Optional: The Date format of date, time, date/time fields in this Lake field. **This field is deprecated in API version 55.0 and later.**
externalName	string	The external name of this field.
isEventDate	boolean	When true, this field contains the event date for behavioral model area objects that are used to partition data.
primaryIndexOrder	int	If supplied, indicates that this field is part of the primary key. The number value (starting at 1) indicates the order of attributes if it's a compound primary key.
isInternalOrganization	boolean	When true, this field contains the value for internal organization. In this case, the value of the field is the name of the internal organization. Landing Objects don't have access to the Salesforce ID and thus are using the developer name.
isRecordModified	boolean	Indicates the record modified field used to calibrate latest record version.
mktDatalakeSrcKeyQualifier	string	String storing the developer name of MktDataLakeSrcKeyQualifier configured on the field. Available in API version 55.0 and later.
keyQualifierName	string	Contains the developer name of key qualifier field. Available in API version 55.0 and later.

LookupFilter

Represents the metadata associated with a lookup filter. Replaces the NamedFilter component, which was removed as of API version 30.0. LookupFilter is available in API version 30.0 and later.

Field	Field Type	Description
`active`	boolean	Required. Indicates whether the lookup filter is active (`true`) or not (`false`).
`booleanFilter`	string	Specifies advanced filter conditions.
`description`	string	A description of what this filter does.
`errorMessage`	string	The error message that appears if the lookup filter fails.
`filterItems`	FilterItem[]	Required. The set of filter conditions. You can have up to 10 FilterItems per lookup filter.
`infoMessage`	string	The information message displayed on the page. Use to describe things the user possibly doesn't understand, such as why certain items are excluded in the lookup filter.
`isOptional`	boolean	Required. Indicates whether the lookup filter is optional (`true`) or not (`false`).

Lookup filters use additional data types. For more information, see Metadata Field Types.

FilterItem

Represents one entry in a set of filter criteria.

Field	Field Type	Description
`field`	string	Represents the field specified in the filter.
`operation`	FilterOperation (enumeration of type string)	Represents the filter operation for this filter item. Valid values are: • `equals` • `notEqual` • `lessThan` • `greaterThan` • `lessOrEqual` • `greaterOrEqual` • `contains` • `notContain` • `startsWith` • `includes` • `excludes` • `within` (`DISTANCE` criteria only)
`value`	string	Represents the value of the filter item being operated upon, for example, if the filter is `my_number_field__c > 1`, the value of `value` is `1`.

Field	Field Type	Description
valueField	string	Specifies if the final column in the filter contains a field or a field value. Approval processes don't support valueField entries in filter criteria.

Declarative Metadata Sample Definition

The following example shows a field definition for a custom field that's named Comments__c.

```
<?xml version="1.0" encoding="UTF-8"?>
<CustomObject xmlns="http://soap.sforce.com/2006/04/metadata">
....
<fields>
        <fullName>Comments__c</fullName>
        <description>Add your comments about this object here</description>
        <inlineHelpText>This field contains help text for this object</inlineHelpText>
        <label>Comments</label>
        <length>32000</length>
        <type>LongTextArea</type>
        <visibleLines>30</visibleLines>
</fields>
....
</CustomObject>
```

This XML is the definition for two fields on the Account standard object—a custom field (MyCustomAccountField__c), and a standard field (Phone) that has history tracking enabled.

```
<?xml version="1.0" encoding="UTF-8"?>
<CustomObject xmlns="http://soap.sforce.com/2006/04/metadata">
    <fields>
        <fullName>MyCustomAccountField__c</fullName>
        <description>A custom field on the Account standard object.</description>
        <externalId>false</externalId>
        <inlineHelpText>Some help text.</inlineHelpText>
        <label>MyCustomAccountField</label>
        <length>100</length>
        <required>false</required>
        <trackFeedHistory>false</trackFeedHistory>
        <trackHistory>false</trackHistory>
        <type>Text</type>
        <unique>false</unique>
    </fields>
    <fields>
        <fullName>Phone</fullName>
        <trackFeedHistory>false</trackFeedHistory>
        <trackHistory>true</trackHistory>
    </fields>
</CustomObject>
```

Wildcard Support in the Manifest File

This metadata type doesn't support the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SEE ALSO:

- CustomObject
- Picklist (Including Dependent Picklist)
- Metadata
- NamedFilter

FieldSet

Represents a field set. A field set is a grouping of fields. For example, you could have a field set that contains fields describing a user's first name, middle name, last name, and business title.

Field sets can be referenced on Visualforce pages dynamically. If the page is added to a managed package, administrators can add, remove, or reorder fields in a field set to modify the fields presented on the Visualforce page without modifying any code.

Version

FieldSet components are available in API version 21.0 and later.

Fields

Field	Field Type	Description
`availableFields`	FieldSetItem[]	An array containing all the possible fields in the field set.
`description`	string	Required. A description provided by the developer that describes the field set. This is required.
`displayedFields`	FieldSetItem[]	An array containing all the fields that are presented on the Visualforce page. The order in which a field is listed determines the order of appearance on the page.
`label`	string	Required. The label used to reference the field set.

FieldSetItem

FieldSetItem represents an individual field in a field set.

Field	Field Type	Description
`field`	string	Required. The name of a field in a standard or custom object.
`isFieldManaged`	boolean	Read-only. Denotes whether the field was added to the field set via a managed or unmanaged package.

Field	Field Type	Description
isRequired	boolean	Read-only. Indicates whether the field is universally required (true) or not (false).

Declarative Metadata Sample Definition

A sample XML definition of a FieldSet component is shown below.

```
<?xml version="1.0" encoding="UTF-8"?>
<CustomObject xmlns="http://soap.sforce.com/2006/04/metadata">
    <fieldSets>
        <fullName>FieldSetNames</fullName>
        <availableFields>
            <field>MiddleName__c</field>
        </availableFields>
        <availableFields>
            <field>Title__c</field>
        </availableFields>
        <description>FieldSet containing how to properly address someone</description>
        <displayedFields>
            <field>FirstName__c</field>
        </displayedFields>
        <displayedFields>
            <field>LastName__c</field>
        </displayedFields>
        <label>FieldSet Names</label>
    </fieldSets>
</CustomObject>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character * (asterisk) in the package.xml manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

HistoryRetentionPolicy

Represents the policy for archiving field history data. When you set a policy, you specify the number of months that you want to keep field history in Salesforce before archiving it. By default, when Field Audit Trail is enabled, all field history is retained.

This component is only available to users with the RetainFieldHistory permission.

Declarative Metadata File Suffix and Directory Location

Field history retention policies are defined as part of a standard or custom object. You can set field history retention policies for objects individually. See CustomObject for more information.

Version

Available in API version 31.0 and later.

Fields

Field Name	Field Type	Description
`archiveAfterMonths`	int	Required. The number of months that you want to keep field history data in Salesforce before archiving. You can set a minimum of 1 month and a maximum of 18 months. If you don't set a number, the default is 18 months. (That is, Salesforce maintains data for 18 months before archiving.)
`archiveRetentionYears`	int	The number of years until you manually delete data from the archive. Use this field as a reminder for manually deleting data. By default, field history data isn't automatically deleted when Field Audit Trail is enabled.
`description`	string	A text description for the history retention.
`gracePeriodDays`	int	The number of days of extra time after the `archiveAfterMonths` period before the data is archived. The `gracePeriodDays` interval applies only to the first time that the data is archived; because all the data is copied the first time, the operation can take longer than subsequent times when only the data that changed since the last archival operation is copied. The `gracePeriodDays` provides extra time for the administrator to prepare the organization before the initial archive operation. You can set a minimum of zero days and a maximum of 10 days. If no number is set, the default is 1 day.

Declarative Metadata Sample Definition

This sample shows the definition of a history retention policy for a custom object.

```
<?xml version="1.0" encoding="UTF-8"?>
<CustomObject xmlns="http://soap.sforce.com/2006/04/metadata">
   <historyRetentionPolicy>
       <archiveAfterMonths>6</archiveAfterMonths>
       <archiveRetentionYears>5</archiveRetentionYears>
       <description>My field history retention</description>
   </historyRetentionPolicy>

...
</CustomObject>
```

Index

Represents an index defined within a custom big object. Use this metadata type to define the composite primary key (index) for a custom big object.This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

Indexes are user-defined and are part of the custom object definition for big objects. See CustomObject for more information.

Version

The Index type is available in API version 41.0 and later.

Fields

Field Name	Field Type	Description
`fields`	IndexField[]	The definition of the fields in the index.
`label`	string	Required. This name is used to refer to the big object in the user interface. Available in API version 41.0 and later.

IndexField

Defines which fields make up the index, their order, and sort direction. The order in which the fields are defined determines the order fields are listed in the index.

Field Name	Field Type	Description
`name`	string	Required. The API name for the field that's part of the index. This value must match the `fullName` value for the corresponding field in the fields section and be marked as required. **Warning:** When querying a big object record via SOQL and passing the results as arguments to the delete API, if any index field name has a leading or trailing white space, you can't delete the big object record.
`sortDirection`	string	Required. The sort direction of the field in the index. Valid values are `ASC` for ascending order and `DESC` for descending order.

Declarative Metadata Sample Definition

The following is an example of an index contained within the definition of a custom big object, `Customer_Interactions__b.object`.

```
<?xml version="1.0" encoding="UTF-8"?>
<CustomObject xmlns="http://soap.sforce.com/2006/04/metadata">

    <deploymentStatus>Deployed</deploymentStatus>

// Define the fields within the big object
    <fields>
        <fullName>Purchase__c</fullName>
        <label>Purchase</label>
        <length>16</length>
        <required>false</required>
        <type>Text</type>
        <unique>false</unique>
    </fields>
```

```
    <fields>
        <fullName>Order_Number__c</fullName>
        <label>Order Number</label>
        <length>16</length>
        <required>false</required>
        <type>Text</type>
        <unique>true</unique>
    </fields>

    <fields>
        <fullName>Platform__c</fullName>
        <label>Platform</label>
        <length>16</length>
        <required>true</required>
        <type>Text</type>
        <unique>false</unique>
    </fields>

    <fields>
        <fullName>Account__c</fullName>
        <label>User Account</label>
        <referenceTo>Account</referenceTo>
        <relationshipName>User_Account</relationshipName>
        <required>true</required>
        <type>Lookup</type>
    </fields>

    <fields>
        <fullName>Order_Date__c</fullName>
        <label>Order Date</label>
        <required>true</required>
        <type>DateTime</type>
    </fields>

// Define the index
    <indexes>
        <fullName>CustomerInteractionsIndex</fullName>
        <label>Customer Interactions Index</label>
        <fields>
            <name>Account__c</name>
            <sortDirection>DESC</sortDirection>
        </fields>
        <fields>
            <name>Platform__c</name>
            <sortDirection>ASC</sortDirection>
        </fields>
        <fields>
            <name>Order_Date__c</name>
            <sortDirection>DESC</sortDirection>
        </fields>
    </indexes>

    <label>Customer Interaction</label>
```

```
    <pluralLabel>Customer Interactions</pluralLabel>
</CustomObject>
```

Wildcard Support in the Manifest File

This metadata type doesn't support the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SEE ALSO:

- CustomObject
- Metadata

ListView

ListView allows you to see a filtered list of records, such as contacts, accounts, or custom objects.

This type extends the Metadata metadata type and inherits its `fullName` field. See "Create a Custom List View in Salesforce Classic" in Salesforce Help.

Note: List views with the Visible only to me `Restrict Visibility` option aren't accessible in Metadata API. Each of these list views is associated with a particular user.

Declarative Metadata File Suffix and Directory Location

List views are stored within a CustomObject component. The component can represent a custom object or a standard object, such as an account.

Version

ListView components for custom objects are available in API version 14.0 and later. ListView components for standard objects, such as accounts, are available in API version 17.0 and later.

Fields

Field	Field Type	Description
`booleanFilter`	string	This field represents an Advanced Option for a filter. Advanced Options in filters allow you to build up filtering conditions that use a mixture of AND and OR boolean operators across multiple filter line items. For example, `(1 AND 2) OR 3` finds records that match both the first two filter line items or the third.
`columns`	string[]	The list of fields in the list view. The field name relative to the object name, for example MyCustomField__c, is specified for each custom field. Field names in the ListView columns don't always match their API name counterparts. If person accounts are enabled in your organization, standard fields merged from a contact into an

Field	Field Type	Description
		account start with the `PC_` prefix, while the corresponding API name starts with the `Person` prefix. For example, the ListView column name is `PC_Email` for a corresponding API field name of `PersonEmail`.
`division`	string	If your organization uses divisions to segment data and you've got the "Affected by Divisions" permission, records in the list view must match this division. This field is only available if you're searching all records. This field is available in API version 17.0 and later.
`filterScope`	FilterScope (enumeration of type string)	Required. This field indicates whether you're filtering by owner or viewing all records.
`filters`	ListViewFilter[]	The list of filter line items.
`fullName`	string	Required. Inherited from Metadata, this field is defined in the WSDL for this metadata type. It must be specified when creating, updating, or deleting. See `createMetadata()` to see an example of this field specified for a call.
`label`	string	Required. The list view name.
`language`	Language	The language used for filtering if your organization uses the Translation Workbench and you're using the `startsWith` or `contains` operator. The values entered as search terms must be in the same language as the filter language. For a list of valid language values, see Language. This field is available in API version 17.0 and later.
`queue`	string	The name of a queue. Objects are sometimes assigned to a queue so that the users who have access to the queue can monitor and manage them. When you create a queue, a corresponding list view is automatically created. See "Create Queues" in Salesforce Help.
`sharedTo`	SharedTo	Sharing access for the list view. This field is available in API version 17.0 and later.

ListViewFilter

ListViewFilter represents a filter line item.

Field	Field Type	Description
`filter`	string	Required. Represents the field specified in the filter.

Field	Field Type	Description
`operation`	FilterOperation (enumeration of type string)	Required. The operation used by the filter, such as `equals`. The valid values are: • `equals` • `notEqual` • `lessThan` • `greaterThan` • `lessOrEqual` • `greaterOrEqual` • `contains` • `notContain` • `startsWith` • `includes` • `excludes` • `within` (`DISTANCE` criteria only)
`value`	string	Represents the value of the filter item being operated upon, for example, if the filter is `my_number_field__c > 1`, the value of `value` is `1`.

FilterScope

The FilterScope is an enumeration of type string that represents the filtering criteria for the records. The valid values are listed in the table:

Enumeration Value	Description
`Everything`	All records, for example All Opportunities.
`Mine`	Records owned by the user running the list view, for example My Opportunities.
`MineAndMyGroups`	Records owned by the user running the list view, and records assigned to the user's queues.
`AssignedToMe`	Records assigned to the user running the list view. The `AssignedToMe` scope is supported for the ServiceAppointment object only.
`Queue`	Records assigned to a queue.
`Delegated`	Records delegated to another user for action: for example, a delegated task. This option is available in API version 17.0 and later.
`MyTerritory`	Records in the territory of the user seeing the list view. This option is available if territory management is enabled for your organization. Opportunities can't be filtered by `MyTerritory`. This option is available in API version 17.0 and later.

Enumeration Value	Description
`MyTeamTerritory`	Records in the territory of the team of the user seeing the list view. This option is available if territory management is enabled for your organization. Opportunities can't be filtered by `MyTeamTerritory`. This option is available in API version 17.0 and later.
`Team`	Records assigned to a team. In the Lightning Experience UI, the corresponding list view filter is **My team's opportunities**. This option is available in API version 17.0 and later.
`SalesTeam`	Opportunities assigned to an opportunity team. In the Lightning Experience UI, the corresponding list view filter is **My opportunity teams**. This option is available in API version 49.0 and later.
`ScopingRule`	Records that meet a scoping rule's record criteria. In Lightning Experience, scoping rules are applied to list views only if the user selects **Filter by scope**.

Declarative Metadata Sample Definition

A sample XML definition of a list view in a custom object is shown.

```
<?xml version="1.0" encoding="UTF-8"?>
<CustomObject xmlns="http://soap.sforce.com/2006/04/metadata">
. . .
    <listViews>
        <fullName>All_Mileages</fullName>
        <filterScope>everything</filterScope>
        <label>All Mileages</label>
    </listViews>
    <listViews>
        <fullName>My_Mileages</fullName>
        <booleanFilter>1 AND 2</booleanFilter>
        <columns>NAME</columns>
        <columns>CREATED_DATE</columns>
        <filterScope>mine</filterScope>
        <filters>
            <field>NAME</field>
            <operation>equals</operation>
            <value>Eric Bristow</value>
        </filters>
        <filters>
            <field>City__c</field>
            <operation>equals</operation>
            <value>Paris</value>
        </filters>
        <label>My Mileages</label>
    </listViews>
. . .
</CustomObject>
```

Wildcard Support in the Manifest File

This metadata type doesn't support the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SEE ALSO:

- CustomObject
- Sample package.xml Manifest Files

NamedFilter

Represents the metadata associated with a lookup filter. This metadata type is used to create, update, or delete lookup filter definitions. This component has been removed as of API version 30.0 and is only available in previous API versions. The metadata associated with a lookup filter is now represented by the lookupFilter field in the CustomField component.

This type extends the Metadata metadata type and inherits its `fullName` field. You can also use this metadata type to work with customizations of lookup filters on standard fields.

Note: The namedFilter appears as a child of the target object of the associated lookup field.

Declarative Metadata File Suffix and Directory Location

Lookup filters are defined as part of the custom object or standard object definition. See CustomObject for more information.

Note: Retrieving a component of this metadata type in a project makes the component appear in any Profile and PermissionSet components that are retrieved in the same package.

Version

Lookup filters are available in API version 17.0 and later. However, the NamedFilter type was removed in API version 30.0. The metadata associated with a lookup filter is now represented by the lookupFilter field in the CustomField type.

Fields

Unless otherwise noted, all fields are creatable, filterable, and nillable.

Field Name	Field Type	Description
`active`	boolean	Required. Indicates whether the lookup filter is active.
`booleanFilter`	string	Specifies advanced filter conditions.
`description`	string	A description of what this filter does.
`errorMessage`	string	The error message that appears if the lookup filter fails.

Field Name	Field Type	Description
`field`	string	Required. The `fullName` of the custom or standard field associated with the lookup filter. You can associate one relationship field with each lookup filter, and vice versa. Note: You can't update a field associated with a lookup filter.
`filterItems`	FilterItems[]	Required. The set of filter conditions.
`infoMessage`	string	The information message displayed on the page. Use to describe things the user might not understand, such as why certain items are excluded in the lookup filter.
`fullName`	string	Inherited from Metadata, this field is defined in the WSDL for this metadata type. It must be specified when creating, updating, or deleting. See `createMetadata()` to see an example of this field specified for a call. This value can't be `null`.
`isOptional`	boolean	Required. Indicates whether the lookup filter is optional.
`name`	string	Required. The name of the lookup filter. If you create this field in the user interface, a name is automatically assigned. If you create this field through Metadata API, you must include the `name` field.
`sourceObject`	string	The object that contains the lookup field that uses this lookup filter. Set this field if the lookup filter references fields on the source object.

Lookup filters use additional data types. For more information, see Metadata Field Types.

FilterItems

FilterItems contains the following properties:

Field	Field Type	Description
`field`	string	Represents the field specified in the filter.
`operation`	FilterOperation (enumeration of type string)	Represents the filter operation for this filter item. Valid values are enumerated in FilterOperation.
`value`	string	Represents the value of the filter item being operated upon, for example, if the filter is `my_number_field__c > 1`, the value of `value` is `1`.

FilterOperation

Here's an enumeration of type string that lists different filter operations. Valid values are:

- `equals`
- `notEqual`
- `lessThan`
- `greaterThan`
- `lessOrEqual`
- `greaterOrEqual`
- `contains`
- `notContain`
- `startsWith`
- `includes`
- `excludes`

Declarative Metadata Sample Definition

```
<?xml version="1.0" encoding="UTF-8"?>
<CustomObject xmlns="http://soap.sforce.com/2006/04/metadata">
....
    <namedfilters>
        <fullName>nf_Acc</fullName>
        <active>true</active>
        <booleanFilter>1 OR 2</booleanFilter>
        <field>Account.lk__c</field>
        <filterItems>
            <field>Account.Phone</field>
            <operation>notEqual</operation>
            <value>x</value>
        </filterItems>
        <filterItems>
            <field>Account.Fax</field>
            <operation>notEqual</operation>
            <value>y</value>
        </filterItems>
        <name>Acc</name>
        <sourceObject>Account</sourceObject>
    </namedfilters>
....
</CustomObject>
```

Wildcard Support in the Manifest File

This metadata type doesn't support the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SEE ALSO:

- CustomObject
- Picklist (Including Dependent Picklist)
- Metadata
- CustomField

Picklist (Including Dependent Picklist)

Deprecated. Represents a picklist (or dependent picklist) definition for a custom field in a custom object or a custom or standard field in a standard object, such as an account.

Version

Use this type in API version 37.0 and earlier only. Picklists for custom fields in custom objects are available in API version 12.0 and later. Picklists for custom or standard fields in standard objects, such as accounts, are available in API version 16.0 and later.

In API version 38.0 and later, Picklist is replaced by ValueSet on page 732 on the CustomField type.

Declarative Metadata File Suffix and Directory Location

Picklist definitions are included in the custom object and field with which they're associated.

Fields

Picklist contains the following fields:

Field Name	Field Type	Description
`controllingField`	string	The `fullName` of the controlling field if `controllingField` is a dependent picklist. A dependent picklist works in conjunction with a controlling picklist or checkbox to filter the available options. The value chosen in the controlling field affects the values available in the dependent field. This field is available in API version 14.0 and later.
`picklistValues`	PicklistValue[]	Required. Represents a set of values for a picklist.
`restrictedPicklist`	boolean	Indicates whether the picklist's value list is restricted. With a restricted picklist, only an admin can add or change values; users can't load or remove values through the API. By default this value is `false`. This field is available in API version 37.0 and later.
`sorted`	boolean	Indicates whether values are sorted (`true`), or not (`false`). By default this value is `false`.

Java Sample

The following sample uses a picklist. For a complete sample of using a picklist with record types and profiles, see Profile on page 1616.

```
public void setPicklistValues() {
  // Create a picklist
  Picklist expenseStatus = new Picklist();
  PicklistValue unsubmitted = new PicklistValue();
  unsubmitted.setFullName("Unsubmitted");
  PicklistValue submitted = new PicklistValue();
  submitted.setFullName("Submitted");
  PicklistValue approved = new PicklistValue();
  approved.setFullName("Approved");
  PicklistValue rejected = new PicklistValue();
  rejected.setFullName("Rejected");
  expenseStatus.setPicklistValues(new PicklistValue[]
      {unsubmitted, submitted, approved, rejected});

  CustomField expenseStatusField = new CustomField();
  expenseStatusField.setFullName(
      "ExpenseReport__c.ExpenseStatus__c");
  expenseStatusField.setLabel("Expense Report Status");
  expenseStatusField.setType(FieldType.Picklist);
  expenseStatusField.setPicklist(expenseStatus);
  try {
    AsyncResult[] ars =
    metadataConnection.create(new Metadata[] {expenseStatusField});
  } catch (ConnectionException ce) {
    ce.printStackTrace();
  }
}
```

Declarative Metadata Sample Definition

The following sample shows usage for picklists, including dependent picklists, in a custom object. The `isAmerican__c` checkbox controls the list of manufacturers shown in the `manufacturer__c` picklist. The `manufacturer__c` checkbox in turn controls the list of models shown in the `model__c` picklist.

```
<?xml version="1.0" encoding="UTF-8"?>
<CustomObject xmlns="http://soap.sforce.com/2006/04/metadata">
    <deploymentStatus>Deployed</deploymentStatus>
    <enableActivities>true</enableActivities>
    <fields>
        <fullName>isAmerican__c</fullName>
        <defaultValue>false</defaultValue>
        <label>American Only</label>
        <type>Checkbox</type>
    </fields>
    <fields>
        <fullName>manufacturer__c</fullName>
        <label>Manufacturer</label>
        <picklist>
            <controllingField>isAmerican__c</controllingField>
            <picklistValues>
```

```
            <fullName>Chrysler</fullName>
            <controllingFieldValues>checked</controllingFieldValues>
            <default>false</default>
        </picklistValues>
        <picklistValues>
            <fullName>Ford</fullName>
            <controllingFieldValues>checked</controllingFieldValues>
            <default>false</default>
        </picklistValues>
        <picklistValues>
            <fullName>Honda</fullName>
            <controllingFieldValues>unchecked</controllingFieldValues>
            <default>false</default>
        </picklistValues>
        <picklistValues>
            <fullName>Toyota</fullName>
            <controllingFieldValues>unchecked</controllingFieldValues>
            <default>false</default>
        </picklistValues>
        <sorted>false</sorted>
    </picklist>
    <type>Picklist</type>
</fields>
<fields>
    <fullName>model__c</fullName>
    <label>Model</label>
    <picklist>
        <controllingField>manufacturer__c</controllingField>
        <picklistValues>
            <fullName>Mustang</fullName>
            <controllingFieldValues>Ford</controllingFieldValues>
            <default>false</default>
        </picklistValues>
        <picklistValues>
            <fullName>Taurus</fullName>
            <controllingFieldValues>Ford</controllingFieldValues>
            <default>false</default>
        </picklistValues>
        <picklistValues>
            <fullName>PT Cruiser</fullName>
            <controllingFieldValues>Chrysler</controllingFieldValues>
            <default>false</default>
        </picklistValues>
        <picklistValues>
            <fullName>Pacifica</fullName>
            <controllingFieldValues>Chrysler</controllingFieldValues>
            <default>false</default>
        </picklistValues>
        <picklistValues>
            <fullName>Accord</fullName>
            <controllingFieldValues>Honda</controllingFieldValues>
            <default>false</default>
        </picklistValues>
        <picklistValues>
```

```
                <fullName>Civic</fullName>
                <controllingFieldValues>Honda</controllingFieldValues>
                <default>false</default>
            </picklistValues>
            <picklistValues>
                <fullName>Prius</fullName>
                <controllingFieldValues>Toyota</controllingFieldValues>
                <default>false</default>
            </picklistValues>
            <picklistValues>
                <fullName>Camry</fullName>
                <controllingFieldValues>Toyota</controllingFieldValues>
                <default>false</default>
            </picklistValues>
            <sorted>false</sorted>
        </picklist>
        <type>Picklist</type>
    </fields>
....
</CustomObject>
```

The following sample shows usage for the standard `Stage` field in opportunities.

```
<?xml version="1.0" encoding="UTF-8"?>
<CustomObject xmlns="http://soap.sforce.com/2006/04/metadata">
    <fields>
        <fullName>StageName</fullName>
        <picklist>
            <picklistValues>
                <fullName>Prospecting</fullName>
                <default>false</default>
                <forecastCategory>Pipeline</forecastCategory>
                <probability>10</probability>
            </picklistValues>
            <picklistValues>
                <fullName>Qualification</fullName>
                <default>false</default>
                <forecastCategory>Pipeline</forecastCategory>
                <probability>10</probability>
            </picklistValues>
            <picklistValues>
                <fullName>Needs Analysis</fullName>
                <default>false</default>
                <forecastCategory>Pipeline</forecastCategory>
                <probability>20</probability>
            </picklistValues>
            ...
        </picklist>
    </fields>
<CustomObject>
```

Wildcard Support in the Manifest File

This metadata type doesn't support the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ProfileSearchLayouts

Represents a user profile's search results layouts for an object. `ProfileSearchLayouts` are similar to `SearchLayouts`. However, with profile-specific layouts, each user profile can have a different search results layout for an object.

File Suffix and Directory Location

Profile search layouts are defined as part of a standard or custom object. `SearchLayout` is the default search results layout used when no layout is specified for a user profile. For more information, see CustomObject.

Version

Profile search layouts for custom objects are available in API version 48.0 and later.

Fields

Field	Field Type	Description
`profileName`	string[]	The name of the profile associated with a customized search results layout. The profile name can be a standard Salesforce profile or custom profile defined in your org.
`fields`	string[]	The list of fields displayed in search results for the object and for the users that have the profile *`Profile Name`*. The `name` field is required and is always displayed as the first column header, so it isn't included in this list. All additional fields are included. The field name relative to the object name, for *`exampleMyCustomField__c`*, is specified for each custom field.

Declarative Metadata Sample Definition

The following shows a sample definition of profile-specific search layouts in an object.

Note: To deploy a profile-specific search results layout, the profile must be defined in the destination org and if it's for a custom object, you must enable search for that custom object. If the profile-specific search results layout is for a custom object, the custom object's tab must exist in the destination org or must be included with the deployment.

```
<?xml version="1.0" encoding="UTF-8"?>
                <CustomObject xmlns="http://soap.sforce.com/2006/04/metadata">
                . . .
                <profileSearchLayouts>
                <fields>ACCOUNT.NAME</fields>
                <fields>ACCOUNT.SITE</fields>
```

```
<fields>ACCOUNT.PHONE1</fields>
<fields>CORE.USERS.ALIAS</fields>
<fields>ACCOUNT.ADDRESS2_CITY</fields>
<profileName>System Administrator</profileName>
</profileSearchLayouts>
<profileSearchLayouts>
<fields>ACCOUNT.NAME</fields>
<fields>ACCOUNT.SITE</fields>
<profileName>WDC Only User</profileName>
</profileSearchLayouts>
. . .
</CustomObject>
```

SEE ALSO:

SearchLayouts

RecordType

Represents the metadata associated with a record type. Record types let you offer different business processes, picklist values, and page layouts to different users. Use this metadata type to create, update, or delete record type definitions for a custom object.

For more information, see *Tailor Busines Processes to Different Record Types Users* in Salesforce Help. This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Don't use record types as an access control mechanism. Profile assignment governs create and edit access for an object but doesn't govern read access. For example, a user assigned to a profile that isn't enabled for a particular record type can't create records with that record type, but can access records associated with that record type.

Users with access to an object can read all record type information for that object. We strongly recommend against storing sensitive information in the record type description, name, or label. Instead, store sensitive information in a separate object or fields to which you've applied appropriate access controls.

Note: Retrieving a component of this metadata type in a project makes the component appear in any Profile and PermissionSet components that are retrieved in the same package.

Note: Metadata API doesn't retrieve custom picklist values on person account record types, if the picklist exists on a contact. In this case, Metadata API retrieves standard picklist values only.

Version

Record types are available in API version 12.0 and later.

Fields

Field	Field Type	Description
`active`	boolean	Required. Indicates whether the record type is active.
`businessProcess`	string	The `fullName` of the business process associated with the record type. This field is required in record types for lead,

Field	Field Type	Description
		opportunity, solution, and case, and not allowed otherwise. See BusinessProcess on page 682. This field is available in API version 17.0 and later.
`compactLayoutAssignment`	string	Represents the compact layout that is assigned to the record type. This field is available in API version 29.0 and later.
`description`	string	Record type description. Maximum of 255 characters.
`fullName`	string	Record type name. The `fullName` can contain only underscores and alphanumeric characters. It must be unique, begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. If this field contained characters before version 14.0 that are no longer allowed, the characters were stripped out of this field, and the previous value of the field was saved in the `label` field. Inherited from the Metadata component, this field isn't defined in the WSDL for this component. It must be specified when creating, updating, or deleting. See create() to see an example of this field specified for a call. This value can't be `null`.
`label`	string	Required. Descriptive label for the record type. The list of characters allowed in the `fullName` field has been reduced for versions 14.0 and later. This field contains the value contained in the `fullName` field before version 14.0.
`picklistValues`	RecordTypePicklistValue[]	Represents a set of values for a picklist.

RecordTypePicklistValue

RecordTypePicklistValue represents the combination of picklists and valid values that define a record type:

Field Name	Field Type	Description
`picklist`	string	Required. The name of the picklist.
`values`	PicklistValue	One or more of the picklist values in the picklist. Each value defined is available in the record type that contains this component.

Java Sample

The following sample uses two record types. For the complete sample that includes profiles and picklists, see Profile on page 1616.

```
public void recordTypeSample() {
  try {
    // Employees and managers have different access
    // to the state of the expense sheet
    RecordType edit = new RecordType();
    edit.setFullName("ExpenseReport__c.Edit");
    edit.setLabel("ExpenseReport__c.Label");
    PicklistValue unsubmitted = new PicklistValue();
    unsubmitted.setFullName("Unsubmitted");
    PicklistValue submitted = new PicklistValue();
    submitted.setFullName("Submitted");
    RecordTypePicklistValue editStatuses =
        new RecordTypePicklistValue();
    editStatuses.setPicklist("ExpenseStatus__c");
    editStatuses.setValues(
        new PicklistValue[] {unsubmitted, submitted});
    edit.setPicklistValues(
        new RecordTypePicklistValue[] {editStatuses});
    AsyncResult[] arsEdit =
        metadataConnection.create(new Metadata[] {edit});

    RecordType approve = new RecordType();
    approve.setFullName("ExpenseReport__c.Approve");
    PicklistValue approved = new PicklistValue();
    approved.setFullName("Approved");
    PicklistValue rejected = new PicklistValue();
    rejected.setFullName("Rejected");
    RecordTypePicklistValue approveStatuses =
        new RecordTypePicklistValue();
    approveStatuses.setPicklist("ExpenseStatus__c");
    approveStatuses.setValues(
        new PicklistValue[] {approved, rejected});
    approve.setPicklistValues(
        new RecordTypePicklistValue[] {approveStatuses});
    AsyncResult[] arsApprove =
      metadataConnection.create(new Metadata[] {approve});
  } catch (ConnectionException ce) {
    ce.printStackTrace();
  }
}
```

Declarative Metadata Sample Definition

The definition of a record type in a custom object is shown in this code block.

```
<CustomObject xmlns="http://soap.sforce.com/2006/04/metadata">
. . .
   <recordTypes>
        <fullName>My First Recordtype</fullName>
```

```
    </recordTypes>
 . . .
</CustomObject>
```

Wildcard Support in the Manifest File

This metadata type doesn't support the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SearchLayouts

Represents the metadata associated with the search layouts for an object. You can customize which fields to display for users in search results, search filter fields, lookup dialogs, and recent record lists on tab home pages. You can access SearchLayouts only by accessing its encompassing CustomObject.

For more information, see Customize Layouts for Search Results and Customize Search Layouts for Custom Objects in Salesforce Help.

Version

Search layouts for custom objects are available in API version 14.0 and later. The ability to modify search layouts for standard objects (except events and tasks) is available in API version 27.0 and later.

Fields

When defining metadata for search layouts:

- Any Name field defined as a text type is mandatory; it's always displayed as the first column in the search results page. When you query for a list of fields; the name field isn't returned but all other fields are. If you define the Name field as an autonumber type, it's not mandatory and you can remove it from the list, but when you import the search layout with Metadata API, it will always add the Name field back. These rules apply to `customTabListAdditionalFields`, `lookupDialogsAdditionalFields`, `lookupPhoneDialogsAdditionalFields`, and `searchResultsAdditionalFields`
- For custom objects, the search layout uses the API name, for example, MyCustomField__c instead of the field name My Custom Field.

Field	Field Type	Description
`customTabListAdditionalFields`	string[]	The list of fields displayed in the Recent *`Object Name`* list view for an object.
`excludedStandardButtons`	string[]	The list of standard buttons excluded from the search layout.
`listViewButtons`	string[]	The list of buttons available in list views for an object. This field is equivalent to the Buttons Displayed value in the *`Object Name`* `List View` in the related list of the object detail page in the UI.
`lookupDialogsAdditionalFields`	string[]	The list of fields displayed in a lookup dialog for the object. Salesforce objects often include one or more *lookup fields* that allow users to associate two records together in a relationship. For example, a contact record includes an

Field	Field Type	Description
		`Account` lookup field that represents the relationship between the contact and the organization with which the contact is associated. A lookup search dialog helps you search for the record associated with the one being edited. Lookup filter fields allow you to filter your lookup search by a customized list of fields in the object. This field is equivalent to the `Lookup Dialogs` related list on the object detail page in the UI.
`lookupFilterFields`	string[]	The list of fields that can be used to filter enhanced lookups for an object. Enhanced lookups are optionally enabled by your administrator. This field is equivalent to the `Lookup Filter Fields` related list on the object detail page in the application user interface.
`lookupPhoneDialogsAdditionalFields`	string[]	The list of phone-related fields displayed in a lookup dialog for the object. This list enables integration of the fields with a softphone dial pad. This field is equivalent to the `Lookup Phone Dialogs` related list on the object detail page in the application user interface.
`massQuickActions`	string[]	The list of actions that you can use to perform mass quick action on records. Use this field to add an existing create or update action. You can perform mass quick actions on custom objects and all standard objects that support quick actions and have a search layout in Lightning Experience. This includes but isn't limited to cases, leads, accounts, campaigns, contacts, opportunities, and work orders.
`searchFilterFields`	string[]	The list of fields that can be used to filter a search for the object. This field is equivalent to the `Search Filter Fields` related list on the object detail page in the application user interface.
`searchResultsAdditionalFields`	string[]	The list of fields displayed in a search result for the object. This field is equivalent to the `Search Results` related list on the object detail page in the application user interface.

Field	Field Type	Description
searchResultsCustomButtons	string[]	The list of custom buttons available in a search result for the object. The actions associated with the buttons can be applied to any of the records returned in the search result.

Declarative Metadata Sample Definition

A sample definition of object's search layout is shown..

```
<?xml version="1.0" encoding="UTF-8"?>
                <CustomObject xmlns="http://soap.sforce.com/2006/04/metadata">
                . . .
                <searchLayouts>
                <listViewButtons>New</listViewButtons>
                <listViewButtons>Accept</listViewButtons>
                <listViewButtons>ChangeOwner</listViewButtons>
              <lookupDialogsAdditionalFields>firstQuote__c</lookupDialogsAdditionalFields>

              <lookupDialogsAdditionalFields>finalQuote__c</lookupDialogsAdditionalFields>

                <massQuickActions>Create_MQA_Contact</massQuickActions>
              <searchResultsAdditionalFields>CREATEDBY_USER</searchResultsAdditionalFields>

                </searchLayouts>
                . . .
                </CustomObject>
```

Wildcard Support in the Manifest File

This metadata type doesn't support the wildcard character * (asterisk) in the package.xml manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SEE ALSO:

- CustomObject
- ProfileSearchLayouts

SharingReason

Represents an Apex sharing reason, which is used to indicate why sharing was implemented for a custom object. Apex managed sharing allows developers to use Apex to programmatically share custom objects. When you use Apex managed sharing to share a custom object, only users with the "Modify All Data" permission can add or change the sharing on the custom object's record, and the sharing access is maintained across record owner changes.

Use SharingReason to create, update, or delete sharing reason definitions for a custom object. This type extends the Metadata metadata type and inherits its fullName field.

Version

Sharing reasons are available in API version 14.0 and later.

Fields

Field	Field Type	Description
`fullName`	string	Required. Sharing reason name. The __c suffix is appended to custom sharing reasons. Inherited from Metadata, this field is defined in the WSDL for this metadata type. It must be specified when creating, updating, or deleting. See `createMetadata()` to see an example of this field specified for a call.
`label`	string	Required. Descriptive label for the sharing reason. Maximum of 40 characters.

Declarative Metadata Sample Definition

The definition of a sharing reason in a custom object:

```
<CustomObject xmlns="http://soap.sforce.com/2006/04/metadata">
. . .
    <sharingReasons>
        <fullName>recruiter__c</fullName>
        <label>Recruiter</label>
    </sharingReasons>
 . . .
</CustomObject>
```

Wildcard Support in the Manifest File

This metadata type doesn't support the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SharingRecalculation

Represents Apex classes that recalculate the Apex managed sharing for a specific custom object.

Version

Sharing recalculations are available in API version 14.0 and later.

Fields

Field	Field Type	Description
`className`	string	Required. The Apex class that recalculates the Apex sharing for a custom object. This class must implement the `Database.Batchable` interface.

Declarative Metadata Sample Definition

The definition of a sharing recalculation in a custom object:

```
<CustomObject xmlns="http://soap.sforce.com/2006/04/metadata">
. . .
    <sharingRecalculations>
        <className>RecruiterRecalculation</className>
    </sharingRecalculations>
 . . .
</CustomObject>
```

Wildcard Support in the Manifest File

This metadata type doesn't support the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ValidationRule

Represents a validation rule, which is used to verify that the data a user enters in a record is valid and can be saved. A validation rule contains a formula or expression that evaluates the data in one or more fields and returns a value of `true` or `false`. Validation rules also include an error message that your client application can display to the user when the rule returns a value of `true` due to invalid data.

This type extends the Metadata metadata type and inherits its `fullName` field.

As of API version 20.0, validation rules can't have compound fields. Examples of compound fields include addresses, first and last names, dependent picklists, and dependent lookups.

As of API version 40.0, you can use validation rules with custom metadata types.

Version

Validation rules are available in API version 12.0 and later.

Fields

Field Name	Field Type	Description
`active`	boolean	Required. Indicates whether this validation rule is active, (`true`), or not active (`false`).
`description`	string	A description of the validation rule.
`errorConditionFormula`	string	Required. The formula defined in the validation rule. If the formula returns a value of `true`, an error message is displayed.
`errorDisplayField`	string	The fully specified name of a field in the application. If a value is supplied, the error message appears next to the specified field. If you do not specify a value or the field isn't visible on the page layout, the value changes automatically to `Top of Page`.

Field Name	Field Type	Description
errorMessage	string	Required. The message that appears if the validation rule fails. The message must be 255 characters or less.
fullName	string	The internal name of the object. White spaces and special characters are escaped for validity. The name must: • Contain characters, letters, or the underscore (_) character • Must start with a letter • Can't end with an underscore • Can't contain two consecutive underscore characters. Inherited from the Metadata component, this field isn't defined in the WSDL for this component. It must be specified when creating, updating, or deleting. See create() to see an example of this field specified for a call.

Declarative Metadata Sample Definition

A sample XML definition of a validation rule in a custom object is shown in this code block.

```
<?xml version="1.0" encoding="UTF-8"?>
<CustomObject xmlns="http://soap.sforce.com/2006/04/metadata">
    <deploymentStatus>Deployed</deploymentStatus>
    <fields>
        <fullName>Mommy_Cat__c</fullName>
        <label>Mommy Cat</label>
        <referenceTo>Cat__c</referenceTo>
        <relationshipName>Cats</relationshipName>
        <type>Lookup</type>
    </fields>
    <label>Cat</label>
    <nameField>
        <label>Cat Name</label>
        <type>Text</type>
    </nameField>
    <pluralLabel>Cats</pluralLabel>
    <sharingModel>ReadWrite</sharingModel>
    <validationRules>
        <fullName>CatsRule</fullName>
        <active>true</active>
        <errorConditionFormula>OR(Name = &apos;Milo&apos;,Name =
&apos;Moop&apos;)</errorConditionFormula>
        <validationMessage>Name must be that of one of my cats</validationMessage>
    </validationRules>
</CustomObject>
```

Wildcard Support in the Manifest File

This metadata type doesn't support the wildcard character * (asterisk) in the package.xml manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

WebLink

Represents a custom button or link defined in a custom object.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

This type extends the Metadata metadata type and inherits its `fullName` field.

Version

WebLinks are available in API version 12.0 and later.

Fields

Field Name	Field Type	Description
`availability`	WebLinkAvailability (enumeration of type string)	Required. Indicates whether the button or link is only available online (`online`, or if it is also available offline (`offline`).
`description`	string	A description of the button or link.
`displayType`	WebLinkDisplayType (enumeration of type string)	Represents how the button or link is rendered. Valid values are: • `link` for a hyperlink • `button` for a button • `massActionButton` for a button attached to a related list
`encodingKey`	Encoding	Required. The default encoding setting is Unicode: `UTF-8`. Change it if your template requires data in a different format. This is available if your content source is URL. Valid values include: • `UTF-8`—Unicode (UTF-8) • `ISO-8859-1`—General US & Western Europe (ISO-8859–1, ISO-LATIN-1) • `Shift_JIS`—Japanese (Shift-JIS) • `ISO-2022-JP`—Japanese (JIS) • `EUC-JP`—Japanese (EUC-JP) • `x-SJIS_0213`—Japanese (Shift-JIS_2004) • `ks_c_5601-1987`—Korean (ks_c_5601-1987) • `Big5`—Traditional Chinese (Big5) • `GB2312`—Simplified Chinese (GB2312) • `Big5-HKSCS`—Traditional Chinese Hong Kong (Big5–HKSCS)
`fullName`	string	The name of the custom button or link with white spaces and special characters escaped for validity. The name can only contain characters, letters, and the underscore (_) character. The name must start with a

Field Name	Field Type	Description
		letter, and can't end with an underscore or contain two consecutive underscore characters. Inherited from the Metadata component, this field isn't defined in the WSDL for this component. It must be specified when creating, updating, or deleting. See create() to see an example of this field specified for a call.
`hasMenubar`	boolean	If the `openType` is `newWindow`, this field indicates whether to show the browser menu bar for the window (`true`) or not (`false`). Otherwise, leave this field empty.
`hasScrollbars`	boolean	If the `openType` is `newWindow`, this field indicates whether to show the scroll bars for the window (`true`) or not (`false`). Otherwise, leave this field empty.
`hasToolbar`	boolean	If the `openType` is `newWindow`, this field indicates whether to show the browser toolbar for the window (`true`) or not (`false`). Otherwise, leave this field empty.
`height`	int	Height in pixels of the window opened by the custom button or link. Required if the `openType` is `newWindow`. Otherwise, leave this field empty.
`isResizable`	boolean	If the `openType` is `newWindow`, this field indicates whether to allow resizing of the window (`true`) or not (`false`). Otherwise, leave this field empty.
`linkType`	WebLinkType (enumeration of type string)	Required. Represents whether the content of the button or link is specified by a URL, an sControl, a JavaScript code block, or a Visualforce page. • `url` • `sControl` • `javascript` • `page` • `flow`—Reserved for future use.
`masterLabel`	string	Master label for this object. This display value is the internal label that is not translated.
`openType`	WebLinkWindowType (enumeration of type string)	Required. When the button or link is clicked, specifies the window style that will be used to display the content. Valid values: • `newWindow` • `sidebar` • `noSidebar` • `replace` • `onClickJavaScript`

Field Name	Field Type	Description
page	string	If the value of linkType is page, this field represents the Visualforce page. Otherwise, leave this field empty.
position	WebLinkPosition (enumeration of type string)	If the value of OpenType is newWindow, this field indicates how the new window should be displayed. Otherwise, don't specify a value. Valid values are: • fullScreen • none • topLeft
protected	boolean	Required. Indicates whether this subcomponent is protected (true) or not (false). Protected subcomponents can't be linked to or referenced by components or subcomponents created in the installing organization.
requireRowSelection	boolean	If the displayType is massActionButton, this field indicates whether to require individual row selection to execute the action for this button (true) or not (false). Otherwise, leave this field empty.
scontrol	string	If the value of linkType is sControl, this field represents the name of the sControl. Otherwise, leave this field empty.
showsLocation	boolean	If the openType is newWindow, this field indicates whether to show the browser location bar for the window (true) or not (false). Otherwise, leave this field empty.
showsStatus	boolean	If the openType is newWindow, this field indicates whether to show the browser status bar for the window. Otherwise, leave this field empty.
url	string	If the value of linkType is url, this is the URL value. If the value of linkType is javascript, this is the JavaScript content. If the value is neither of these options, leave this field empty. Content must be escaped in a manner consistent with XML parsing rules.
width	int	Width in pixels of the window opened by the button or link. Required if the openType is newWindow. Otherwise, leave this field empty.

Java Sample

The following Java sample shows sample values for WebLink fields:

```
public void WebLinkSample(String name) throws Exception {
    WebLink WebLink = new WebLink();
    // name variable represents the full name of the object
    // on which to create the WebLink, for example, customObject__c
    WebLink.setFullName(name + ".googleButton");
```

```
    WebLink.setUrl("http://www.google.com");
    WebLink.setAvailability(WebLinkAvailability.online);
    WebLink.setLinkType(WebLinkType.url);
    WebLink.setEncodingKey(Encoding.fromString("UTF-8"));
    WebLink.setOpenType(WebLinkWindowType.newWindow);
    WebLink.setHeight(600);
    WebLink.setWidth(600);
    WebLink.setShowsLocation(false);
    WebLink.setHasScrollbars(true);
    WebLink.setHasToolbar(false);
    WebLink.setHasMenubar(false);
    WebLink.setShowsStatus(false);
    WebLink.setIsResizable(true);
    WebLink.setPosition(WebLinkPosition.none);
    WebLink.setMasterLabel("google");
    WebLink.setDisplayType(WebLinkDisplayType.link);

    AsyncResult[] asyncResults = metadataConnection.create(new WebLink[]{WebLink});
    // After the create() call completes, we must poll the results of checkStatus()
    //

}
```

Declarative Metadata Sample Definition

The following is the definition of a WebLink in a custom object. For related samples, see Declarative Metadata Sample Definition and Declarative Metadata Sample Definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<CustomObject xmlns="http://soap.sforce.com/2006/04/metadata">
....
    <WebLinks>
        <fullName>googleButton</fullName>
        <availability>online</availability>
        <displayType>link</displayType>
        <encodingKey>UTF-8</encodingKey>
        <hasMenubar>false</hasMenubar>
        <hasScrollbars>true</hasScrollbars>
        <hasToolbar>false</hasToolbar>
        <height>600</height>
        <isResizable>true</isResizable>
        <linkType>url</linkType>
        <masterLabel>google</masterLabel>
        <openType>newWindow</openType>
        <position>none</position>
        <protected>false</protected>
        <showsLocation>false</showsLocation>
        <showsStatus>false</showsStatus>
        <url>http://www.google.com</url>
        <width>600</width>
    </WebLinks>
....
</CustomObject>
```

Wildcard Support in the Manifest File

This metadata type doesn't support the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SEE ALSO:

- HomePageComponent
- HomePageLayout
- CustomPageWebLink

Metadata Field Types

These field types extend the field types described in the *Salesforce Object Reference*.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Field Type	Objects	What the Field Contains
CustomField	Custom object Custom field	Represents a custom field.
DeleteConstraint	Custom field	A string that represents deletion options for lookup relationships. Valid values are: • `SetNull` • `Restrict` • `Cascade`
DeploymentStatus	Custom object Custom field	A string which represents the deployment status of a custom object or field. Valid values are: • `InDevelopment` • `Deployed`
FieldType	Custom field	Indicates the type of a custom field. Valid values are: • `Address` (beta) • `AutoNumber` • `Lookup` • `MasterDetail` • `MetadataRelationship` • `Checkbox` • `Currency` • `Date` • `DateTime` • `Email`

Field Type	Objects	What the Field Contains
		• `EncryptedText` Note: This page is about Classic Encryption, not Shield Platform Encryption. What's the difference? • `ExternalLookup` • `IndirectLookup` • `Number`[1] • `Percent` • `Phone` • `Picklist` • `MultiselectPicklist` • `Summary` • `Text` • `TextArea` • `LongTextArea` • `Url` • `Hierarchy` • `File` • `Html` • `Location` (use for geolocation fields) • `Time` [1] A `Number` custom field is internally represented as a field of type double. Setting the scale of the `Number` field to 0 gives you a double that behaves like an int.
Gender	Custom object	Indicates the gender of the noun that represents the object. This is used for languages where words need different treatment depending on their gender. Valid values are: • `Masculine` • `Feminine` • `Neuter` • `AnimateMasculine` (Slavic languages—currently Czech, Polish, Russian, Slovak, Slovenian, and Ukrainian) • `ClassI`, `ClassIII`, `ClassV`, `ClassVII`, `ClassIX`, `ClassXI`, `ClassXIV`, `ClassXV`, `ClassXVI`, `ClassXVII`, `ClassXVIII` (African languages—currently Afrikaans, Xhosa, and Zulu) Note: The following genders are displayed on the Rename Tabs and Labels page in Setup but are stored internally as "Feminine". When setting them through the Metadata API, use "Feminine". • `Euter (Swedish)` • `Common (Dutch)`

Field Type	Objects	What the Field Contains
Picklist (Including Dependent Picklist)	Custom field	(This field type isn't used in Metadata API. CustomField includes this field type for Tooling API support). Represents a picklist, a set of labels and values that can be selected from a picklist.
SharingModel	Custom object	Represents the sharing model for the custom object. Depending on the object, valid values are: • `Private` • `Read` • `ReadWrite` • `ReadWriteTransfer` • `FullAccess` • `ControlledByParent` • `ControlledByCampaign` • `ControlledByLeadOrContact` For example, the User object supports `Private` and `Read` values. Accounts, opportunities, and custom objects support `Private`, `Read` and `ReadWrite` values. Campaign members support `ControlledByCampaign` and `ControlledByLeadOrContact`.
StartsWith	Custom object Custom field	Indicates whether the noun starts with a vowel, consonant, or is a special character. This is used for languages where words need different treatment depending on the first character. Valid values are: • `Consonant` • `Vowel` • `Special (for nouns starting with z, or s plus consonants)`
TreatBlanksAs	Custom field	Indicates how blanks should be treated. Valid values are: • `BlankAsBlank` • `BlankAsZero`
ValueSet	Custom field	Represents a set of values that can be selected from a custom picklist field. Defines the valueSet of a custom picklist field.

ValueSet

Represents a set of values that can be selected from a custom picklist field. Defines the valueSet of a custom picklist field.

Field Type	Field Type	Description
`controllingField`	string	The `fullname` of the controlling field if this is a dependent picklist. A controlling field can be a checkbox or picklist field, but in this case it's a picklist. The controlling picklist filters the available values in the dependent picklist.

Field Type	Field Type	Description
`restricted`	boolean	Whether the picklist's values are limited to only the values defined by a Salesforce admin. Values are `true` or `false`.
`valueSetDefinition`	ValueSetValuesDefinition	Defines value-specific settings for a custom dependent picklist. Indicates whether the value set of the custom picklist field is sorted alphabetically.
`valueSetName`	string	The `masterLabel` of the global value set to be used for this picklist field.
`valueSettings`	ValueSettings	Used for the settings that describe a value in a custom picklist field. The picklist can have its own unique value set, or inherit the values from a global value set. You can add field dependency values via Metadata API but not remove them.

ValueSetValuesDefinition

Field Name	Field Type	Description
`sorted`	boolean	Whether the picklist's value set is displayed in alphabetical order in the user interface.
`value`	CustomValue	Required. The list of values for this local, custom picklist.

ValueSettings

Field Name	Field Type	Description
`controllingFieldValue`	stringstring[]	Applies only to dependent custom picklists. A list of values in the controlling or parent picklist (that the custom picklist values depend on). You can add field dependency values via Metadata API but not remove them.
`valueName`	string	Defines the values in the custom dependent picklist.

CustomObjectTranslation

This metadata type allows you to translate custom objects for a variety of languages.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

This type extends the Metadata metadata type and inherits its `fullName` field. The ability to translate component labels is part of the Translation Workbench.

Declarative Metadata File Suffix and Directory Location

Local translations are stored in a file with a format of *`customObjectName__c-lang`*`.objectTranslation`, where *`customObjectName__c`* is the custom object name, and *`lang`* is the translation language. A sample file name for German translations is `myCustomObject__c-de.objectTranslation`.

Similarly, packaged translations are stored in a file with a format of *customObjectName-pkgNamespace__c-lang*.objectTranslation, where *customObjectName-pkgNamespace__c* is the custom object and package namespace, and *lang* is the translation language. A sample file name for German translations in a package with the namespace of Acme is myCustomObject-Acme__c-de.objectTranslation. Custom object translations are stored in the objectTranslations folder in the corresponding package directory.

Custom object translations are stored in the objectTranslations folder in the corresponding package directory.

Version

CustomObjectTranslation components are available in API version 14.0 and later.

Fields

Field	Field Type	Description
caseValues	ObjectNameCaseValue[]	Different combinations of the custom object with regard to article, plural, possessive, and case.
fields	CustomFieldTranslation[]	A list of translations for the custom fields associated with the custom object.
fieldSets	FieldSetTranslation[]	A list of field set translations. Available in API version 41.0 and later.
fullName	string	The name of the custom object and the translation language with a format of *customObjectName-lang*, where *customObjectName* is the custom object name, and *lang* is the translation language. Inherited from Metadata, this field is defined in the WSDL for this metadata type. It must be specified when creating, updating, or deleting. See createMetadata() to see an example of this field specified for a call.
gender	Gender	Indicates the gender of the noun that represents the object. This is used for languages where words need different treatment depending on their gender.
layouts	LayoutTranslation[]	A list of page layout translations.
nameFieldLabel	string	The label for the name field. Maximum of 80 characters.
namedFilters	NamedFilterTranslation[]	A list of translations for lookup filter error messages associated with the custom object. This field has been removed as of API version 30.0 and is only available in prior versions. The translation metadata associated with a lookup filter is now represented by the lookupFilter field in the CustomFieldTranslation on page 735 subtype.
quickActions	QuickActionTranslation[]	A list of translations for actions.

Field	Field Type	Description
recordTypes	RecordTypeTranslation[]	A list of record type translations.
sharingReasons	SharingReasonTranslation[]	A list of sharing reason translations.
startsWith	StartsWith (enumeration of type string)	Indicates whether the noun starts with a vowel, consonant, or is a special character. This is used for languages where words need different treatment depending on the first character.
validationRules	ValidationRuleTranslation[]	A list of validation rule translations.
webLinks	WebLinkTranslation[]	A list of web link translations.
workflowTasks	WorkflowTaskTranslation[]	A list of workflow task translations.

Note: When you retrieve or deploy translations from a package, the translations from the package might override existing translations. The overridden translations appear in the Rename Tabs and Labels UI until you click **Reset** to restore the translations installed by the latest package.

CustomFieldTranslation

CustomFieldTranslation contains details for a custom field translation. In API versions 37.0 and earlier standard picklist values could be translated with CustomFieldTranslation. In API version 38.0, use StandardValueSetTranslation instead. For more details, see CustomField.

Note: Not every language supports all the possible values for the fields in CustomFieldTranslation. For language-specific supported values, see the fully supported languages and end-user languages appendices.

Field	Field Type	Description
caseValues	ObjectNameCaseValue[]	Different combinations of the custom object with regard to article, plural, possessive, and case. Available in API version 29.0 and later.
description	string	Translation for the custom field description.
gender	Gender	Available in API version 29.0 and later.
help	string	Translation for the text that displays in the field-level help hover text for this field.
label	string	Translation for the label. Maximum of 40 characters.
lookupFilter	LookupFilterTranslation	Represents the translation metadata associated with a lookup filter. This field is available in API version 30.0 and later. LookupFilter isn't supported on the article type object.
name	string	Required. The name of the field relative to the custom object; for example, MyField__c.

Field	Field Type	Description
picklistValues	PicklistValueTranslation[]	List of translations for picklist values. See PicklistValue. Note: "Subject" on the Task object is a text field, not a picklist value. It can't be retrieved via Metadata API. Translations can be provided via the Translation Workbench.
relationshipLabel	string	Translation for a lookup relationship label. A lookup relationship allows a field to be associated with another field. The relationship field allows users to select an option from a list of values defined by the other field. Maximum of 80 characters.
startsWith	StartsWith (enumeration of type string)	Indicates whether the noun starts with a vowel, consonant, or is a special character. Used for languages where words need different treatment depending on the first character. Available in API version 29.0 and later.

FieldSetTranslation

FieldSetTranslation contains details for a field set translation. For more details, see FieldSet. Available in API 41.0 and later.

Field	Field Type	Description
label	string	Required. Translation for the field set label. Maximum of 80 characters.
name	string	Required. The field set name.

LayoutTranslation

LayoutTranslation contains details for a page layout translation. For more details, see Fields.

Field	Field Type	Description
layout	string	Required. The layout name.
layoutType	string	
sections	LayoutSectionTranslation[]	An array of layout section translations.

LayoutSectionTranslation

LayoutSectionTranslation contains details for a page layout section translation. For more details, see LayoutSection.

Field	Field Type	Description
label	string	Required. Translation for the label. Maximum of 765 characters.
section	string	Required. The section name.

LookupFilterTranslation

LookupFilterTranslation shows a translation for a lookup filter error message associated with the custom object. Replaces NamedFilterTranslation.

LookupFilterTranslation is available in API version 30.0 and later.

Field	Field Type	Description
`errorMessage`	string	The error message that appears if the lookup filter fails.
`informationalMessage`	string	The information message displayed on the page. Use to describe things some users don't understand, such as why certain items are excluded in the lookup filter.

NamedFilterTranslation

NamedFilterTranslation has been removed as of API version 30.0 and is only available in previous API versions.

NamedFilterTranslation shows a list of translations for lookup filter error messages associated with the custom object. See NamedFilter for more information.

Field	Field Type	Description
`errorMessage`	string	The error message that appears if the lookup filter fails.
`informationalMessage`	string	The information message displayed on the page. Use to describe things the user doesn't understand, such as why certain items are excluded in the lookup filter.
`name`	string	Required. The name of the lookup filter. If you create this field in the user interface, a name is automatically assigned. If you create this field through Metadata API, you must include the `name` field.

ObjectNameCaseValue

ObjectNameCaseValue supports multiple cases and definitions of the custom object name to allow usage in various grammatical contexts.

Note: Not every language supports all the possible values for the fields in ObjectNameCaseValue. For language-specific supported values, see the fully supported languages and end-user languages appendices.

Field	Field Type	Description
`article`	Article (enumeration of type string)	English has two types of articles: definite (*`the`*) and indefinite (*`a`*, *`an`*). The usage of these articles depends mainly on whether you're referring to any member of a group, or to a specific member of a group. The valid values are: • `Definite` • `Indefinite` • `None`

Field	Field Type	Description
caseType	CaseType (enumeration of type string)	The case of the custom object name. The valid values are: • Ablative • Accusative • Adessive • Allative • Causalfinal • Dative • Delative • Distributive • Elative • Essive • Essiveformal • Genitive • Illative • Inessive • Instrumental • Lative • Locative • Nominative • Objective • Partitive • Prepositional • Subjective • Sublative • Superessive • Termanative • Translative • Vocative
plural	boolean	Indicates whether the value field is plural (true) or singular (false).
possessive	Possessive (enumeration of type string)	The possessive case of a language is a grammatical case used to indicate a relationship of possession. The valid values are: • First • None • Second
value	string	Required. The value or label in this grammatical context.

PicklistValueTranslation

PicklistValueTranslation contains details for translation of a picklist value from a local, custom picklist field. For more details, see Picklist (Including Dependent Picklist).

Field	Field Type	Description
`masterLabel`	string	Required. The picklist value defined on the setup page in the application. Displayed wherever a translated label isn't available.
`translation`	string	Required. Translation for the value.

QuickActionTranslation

QuickActionTranslation contains details for an action label in the user interface. For more information, see QuickAction.

Field	Field Type	Description
`aspect`	string	Identifies which quick action label the translated text belongs to. Use this field only when you want to use different strings for the quick action's field label and informational message. Valid values are `Master` and `InfoMessage`. Available in API version 53.0 and later.
`label`	string	Required. Translation for the label. Maximum of 765 characters.
`name`	string	Required. The quick action name.

RecordTypeTranslation

RecordTypeTranslation contains details for a record type name translation. For more details, see RecordType.

Field	Field Type	Description
`label`	string	Required. Translation for the label. Maximum of 765 characters.
`name`	string	Required. The record type name.
`description`	string	Translation for the record type description. Available in API version 42.0 and later.

SharingReasonTranslation

SharingReasonTranslation contains details for a sharing reason translation. For more details, see SharingReason.

Field	Field Type	Description
`label`	string	Required. Translation for the sharing reason.
`name`	string	Required. The sharing reason name.

ValidationRuleTranslation

ValidationRuleTranslation contains details for a validation rule translation. For more details, see ValidationRule.

Field	Field Type	Description
errorMessage	string	Required. Translation for the error message associated with the validation rule failure.
name	string	Required. The validation rule name.

WebLinkTranslation

WebLinkTranslation contains details for a web link translation. For more details, see WebLink.

Field	Field Type	Description
label	string	Required. Translation for the web link label. Maximum of 765 characters.
name	string	Required. The web link name.

WorkflowTaskTranslation

WorkflowTaskTranslation contains details for a workflow task translation. For more details, see Workflow.

Field	Field Type	Description
description	string	Translation for the workflow task description.
name	string	Required. The workflow task name.
subject	string	Translation for the workflow task subject.

Declarative Metadata Sample Definitions

This sample XML definition shows a CustomObjectTranslation for the Description__c object in German, with one custom field, Summary__c. The name and location of the file containing this definition would be `objectTranslations/Description__c-de.objectTranslation`.

```
<?xml version="1.0" encoding="UTF-8"?>
<CustomObjectTranslation xmlns="http://soap.sforce.com/2006/04/metadata">
    <caseValues>
        <caseType>Nominative</caseType>
        <plural>false</plural>
        <value>Beschreibung</value>
    </caseValues>
    <caseValues>
        <caseType>Nominative</caseType>
        <plural>true</plural>
        <value>Beschreibungen</value>
```

```
    </caseValues>
    <caseValues>
        <caseType>Accusative</caseType>
        <plural>false</plural>
        <value>Beschreibung</value>
    </caseValues>
    <caseValues>
        <caseType>Accusative</caseType>
        <plural>true</plural>
        <value>Beschreibungen</value>
    </caseValues>
    <caseValues>
        <caseType>Genitive</caseType>
        <plural>false</plural>
        <value>Beschreibung</value>
    </caseValues>
    <caseValues>
        <caseType>Genitive</caseType>
        <plural>true</plural>
        <value>Beschreibungen</value>
    </caseValues>
    <caseValues>
        <caseType>Dative</caseType>
        <plural>false</plural>
        <value>Beschreibung</value>
    </caseValues>
    <caseValues>
        <caseType>Dative</caseType>
        <plural>true</plural>
        <value>Beschreibungen</value>
    </caseValues>
    <fields>
        <label>Zusammenfassung</label>
        <name>Summary__c</name>
    </fields>
    <gender>Feminine</gender>
    <nameFieldLabel>Beschreibungen</nameFieldLabel>
</CustomObjectTranslation>
```

This sample XML definition shows a CustomObjectTranslation for the Account object, renaming Account to Client (Kunde) in German. The Account object has one standard field, account_number, and one custom field, Account_Code__c. The name and location of the file containing this definition would be `objectTranslations/Account-de.objectTranslation`.

```
<?xml version="1.0" encoding="UTF-8"?>
<CustomObjectTranslation xmlns="http://soap.sforce.com/2006/04/metadata">
    <caseValues>
        <caseType>Nominative</caseType>
        <plural>false</plural>
        <value>Kunde</value>
    </caseValues>
    <caseValues>
        <caseType>Nominative</caseType>
        <plural>true</plural>
        <value>Kunden</value>
    </caseValues>
```

```
    <caseValues>
        <caseType>Accusative</caseType>
        <plural>false</plural>
        <value>Kunden</value>
    </caseValues>
    <caseValues>
        <caseType>Accusative</caseType>
        <plural>true</plural>
        <value>Kunden</value>
    </caseValues>
    <caseValues>
        <caseType>Genitive</caseType>
        <plural>false</plural>
        <value>Kunden</value>
    </caseValues>
    <caseValues>
        <caseType>Genitive</caseType>
        <plural>true</plural>
        <value>Kunden</value>
    </caseValues>
    <caseValues>
        <caseType>Dative</caseType>
        <plural>false</plural>
        <value>Kunden</value>
    </caseValues>
    <caseValues>
        <caseType>Dative</caseType>
        <plural>true</plural>
        <value>Kunden</value>
    </caseValues>
    <fields>
        <caseValues>
            <caseType>Nominative</caseType>
            <plural>false</plural>
            <value>Kundennummer</value>
        </caseValues>
        <caseValues>
            <caseType>Nominative</caseType>
            <plural>true</plural>
            <value>Kundennummern</value>
        </caseValues>
        <gender>Feminine</gender>
        <name>account_number</name>
    </fields>
    <fields>
        <label>Kunden-Code</label>
        <name>Account_Code__c</name>
    </fields>
    <gender>Masculine</gender>
</CustomObjectTranslation>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SEE ALSO:

- CustomObject
- Translations

CustomPageWebLink

Represents a custom link defined in a home page component.

This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

All other custom links are stored as a WebLink in a CustomObject.

Declarative Metadata File Suffix and Directory Location

There is one file per custom link definition, stored in the `weblinks` folder in the corresponding package directory. The file suffix is `.weblink`.

Version

CustomPageWebLinks are available in API version 13.0 and later.

Fields

Field Name	Field Type	Description
`availability`	WebLinkAvailability (enumeration of type string)	Required. Indicates whether the link is only available online (`online`, or if it is also available offline (`offline`).
`description`	string	A description of the link.
`displayType`	WebLinkDisplayType (enumeration of type string)	Represents how this link is rendered. Valid values: • `link` for a hyperlink • `button` for a button • `massActionButton` for a button attached to a related list

Field Name	Field Type	Description
`encodingKey`	Encoding (enumeration of type string)	Required. The default encoding setting is Unicode: `UTF-8`. Change it if your template requires data in a different format. This is available if your content source is URL. Valid values include: • `UTF-8`—Unicode (UTF-8) • `ISO-8859-1`—General US & Western Europe (ISO-8859–1, ISO-LATIN-1) • `Shift_JIS`—Japanese (Shift-JIS) • `ISO-2022-JP`—Japanese (JIS) • `EUC-JP`—Japanese (EUC-JP) • `x-SJIS_0213`—Japanese (Shift-JIS_2004) • `ks_c_5601-1987`—Korean (ks_c_5601-1987) • `Big5`—Traditional Chinese (Big5) • `GB2312`—Simplified Chinese (GB2312) • `Big5-HKSCS`—Traditional Chinese Hong Kong (Big5–HKSCS)
`fullName`	string	The name used as a unique identifier for API access. The `fullName` can contain only underscores and alphanumeric characters. It must be unique, begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
`hasMenubar`	boolean	If the `openType` is `newWindow`, this field indicates whether to show the browser menu bar for the window (`true` or not (`false`). Otherwise, leave this field empty.
`hasScrollbars`	boolean	If the `openType` is `newWindow`, this field indicates whether to show the scroll bars for the window (`true`) or not (`false`). Otherwise, leave this field empty.
`hasToolbar`	boolean	If the `openType` is `newWindow`, this field indicates whether to show the browser toolbar for the window (`true`) or not (`false`). Otherwise, leave this field empty.
`height`	int	Height in pixels of the window opened by the link. Required if the `openType` is `newWindow`. Otherwise, leave this field empty.
`isResizable`	boolean	If the `openType` is `newWindow`, this field indicates whether to allow resizing of the window (`true`) or not (`false`). Otherwise, leave this field empty.
`linkType`	WebLinkType (enumeration of type string)	Required. Represents whether the content of the button or link is specified by a URL, an sControl, a JavaScript code block, or a Visualforce page. • `url` • `sControl` • `javascript` • `page` • `flow`—Reserved for future use.

Field Name	Field Type	Description
masterLabel	string	The label for the link.
openType	WebLinkWindowType (enumeration of type string)	Required. When the link is clicked, this field specifies the window style used to display the content. Valid values are: • newWindow • sidebar • noSidebar • replace • onClickJavaScript
page	string	If the value of linkType is page, this field represents the Visualforce page. Otherwise, leave this field empty.
position	WebLinkPosition (enumeration of type string)	If the openType is newWindow, this field indicates how the new window should be displayed. Otherwise, leave this field empty. Valid values are: • fullScreen • none • topLeft
protected	boolean	Required. Indicates whether this component is protected (true) or not (false). Protected components cannot be linked to or referenced by components created in the installing organization.
requireRowSelection	boolean	If the openType is massAction, this field indicates whether to require individual row selection to execute the action for this button (true) or not (false). Otherwise, leave this field empty.
scontrol	string	If the value of linkType is sControl, this field represents the name of the sControl. Otherwise, leave this field empty.
showsLocation	boolean	If the openType is newWindow, this field indicates whether or not to show the browser location bar for the window. Otherwise, leave this field empty.
showsStatus	boolean	If the openType is newWindow, this field indicates whether or not to show the browser status bar for the window. Otherwise, leave this field empty.
url	string	If the value of linkType is url, this field represents the URL value. If the value of linkType is javascript, this field represents the JavaScript content. If the value is neither of these, leave this field empty. Content must be escaped in a manner consistent with XML parsing rules.

Field Name	Field Type	Description
`width`	int	Width in pixels of the window opened by the link. Required if the `openType` is `newWindow`. Otherwise, leave this field empty.

Declarative Metadata Sample Definition

The following is the definition of a Weblink. For related samples, see HomePageComponent and HomePageLayout.

```
<?xml version="1.0" encoding="UTF-8"?>
<CustomPageWebLink xmlns="http://soap.sforce.com/2006/04/metadata">
    <availability>online</availability>
    <displayType>button</displayType>
    <encodingKey>UTF-8</encodingKey
    <hasMenubar>false</hasMenubar>
    <hasScrollbars>true</hasScrollbars>
    <hasToolbar>false</hasToolbar>
    <height>600</height>
    <isResizable>true</isResizable>
    <linkType>url</linkType>
    <masterLabel>detailPageButon</masterLabel>
    <openType>newWindow</openType>
    <position>none</position>
    <protected>false</protected>
    <showsLocation>false</showsLocation>
    <showsStatus>false</showsStatus>
    <url>http://google.com</url>
</CustomPageWebLink>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SEE ALSO:

- HomePageComponent
- HomePageLayout
- WebLink

CustomPermission

Represents a permission that grants access to a custom feature. This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

CustomPermission components have the suffix `.customPermission` and are stored in the `customPermissions` folder.

Version

CustomPermission components are available in API version 31.0 and later.

Special Access Rules

As of Summer '20 and later, only users who have one of these permissions can access this object:

- View Setup and Configuration
- Manage Session Permission Set Activations
- Assign Permission Sets

Fields

Field Name	Field Type	Description
`connectedApp`	string	The name of the connected app that's associated with this permission. Limit: 80 characters.
`description`	string	The custom permission description. Limit: 255 characters.
`isLicensed`	boolean	Required. Read-only. Indicates whether the appropriate Salesforce license is required before accessing the permission (`true`) or not (`false`).
`label`	string	Required. The custom permission label. Limit: 80 characters.
`requiredPermission`	CustomPermissionDependencyRequired[]	Indicates which custom permissions are required by the parent custom permission. This field is available in API version 32.0 and later.

CustomPermissionDependencyRequired

CustomPermissionDependencyRequired determines whether a custom permission is required by the parent custom permission. A required custom permission must be enabled when its parent is enabled.

Field Name	Field Type	Description
`customPermission`	string	Required. The custom permission name.

Field Name	Field Type	Description
dependency	boolean	Required. Indicates whether this custom permission is required by the parent custom permission (`true`) or not (`false`).

Declarative Metadata Sample Definition

The following is an example of a CustomPermission component.

```
<?xml version="1.0" encoding="UTF-8"?>
<CustomPermission xmlns="http://soap.sforce.com/2006/04/metadata">
   <connectedApp>Acme</connectedApp>
   <description>Read and edit access for Acme accounts.</description>
   <label>Acme Account Full Access</label>
   <requiredPermission>
      <customPermission>Acme_Account_Read</customPermission>
      <dependency>true</dependency>
   </requiredPermission>
</CustomPermission>
```

The following is an example `package.xml` that references the previous definition, as well as other custom permissions that are associated with a connected app.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
   <types>
      <members>Acme</members>
      <name>ConnectedApp</name>
   </types>
   <types>
      <members>Acme_Account_Email_Read</members>
      <members>Acme_Account_Phone_Edit</members>
      <members>Acme_Account_Full_Access</members>
      <members>Acme_Account_Read</members>
      <name>CustomPermission</name>
   </types>
   <types>
      <members>Acme_Account_Email_Read</members>
      <members>Acme_Account_Phone_Edit</members>
      <members>Acme_Account_Full_Access</members>
      <members>Acme_Account_Read</members>
      <name>PermissionSet</name>
   </types>
   <version>64.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

CustomSite

Represents a Salesforce site. Create public websites and applications that are directly integrated with your Salesforce organization, but don't require users to log in with a username and password.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

For more information, see “Salesforce Sites” in Salesforce Help.

Note: CustomSite doesn’t currently support syndication feeds.

This type extends the Metadata metadata type and inherits its `fullName` field.

Declarative Metadata File Suffix and Directory Location

Lightning Platform CustomSite components are stored in the `sites` directory of the corresponding package directory. The file name matches the site name, and the extension is `.site`.

Version

Lightning Platform CustomSite components are available in API version 14.0 and later.

Fields

Field	Field Type	Description
`active`	boolean	Required. Determines whether the site is active.
`allowHomePage`	boolean	Required. Determines whether the standard home page is visible to public users. This field is available in API version 15.0 and later.
`allowStandardAnswersPages`	boolean	Determines whether the standard answer pages are visible to public users. This field is available in API version 19.0 and later.
`allowStandardIdeasPages`	boolean	Required. Determines whether the standard Ideas pages are visible to public users. This field is available in API version 15.0 and later.
`allowStandardLookups`	boolean	Required. Determines whether the standard lookup pages are visible to public users. This field is available in API version 15.0 and later.
`allowStandardPortalPages`	boolean	Required. When enabled, authenticated users in this site can access standard Salesforce pages as allowed by their access controls. When disabled, authenticated users in this site can't access standard Salesforce pages, even if their access controls allow it. If your site serves only Visualforce pages, disabling this setting

Field	Field Type	Description
		helps add a layer of access protection to your site. This field is available in API version 39.0 and later.
`allowStandardSearch`	boolean	Required. Determines whether the standard search pages are visible to public users. This field is available in API version 15.0 and later.
`analyticsTrackingCode`	string	The tracking code associated with your site. Services such as Google Analytics can use this code to track page request data for your site. This field is available in API version 17.0 and later.
`authorizationRequiredPage`	string	The name of the Visualforce page to be displayed when the guest user tries to access a page for which they are not authorized.
`bandwidthExceededPage`	string	The name of the Visualforce page to be displayed when the site has exceeded its bandwidth quota.
`browserXssProtection`	boolean	Required. Determines whether protection against reflected cross-site scripting attacks is enabled. If a reflected cross-site scripting attack is detected, the browser shows a blank page with no content. Available in API version 41.0 and later.
`cachePublicVisualforcePagesInProxyServers`	boolean	Indicates whether proxy servers cache this site's publicly available pages only for unauthenticated guest users (`true`) or not (`false`). When this field is `false`, this site's cache-enabled Visualforce pages are cached in the web browser for both authenticated and unauthenticated users. The default is `true`. See Configure Site Caching in Salesforce Help for more information. This field is available in API version 52.0 and later.
`changePasswordPage`	string	The name of the Visualforce page to be displayed when the portal user attempts to change their password for either the portal or for Chatter Answers, when enabled.
`chatterAnswersForgotPasswordConfirmPage`	string	The name of the Visualforce page to be displayed that informs the user that an email has been sent to them with a temporary password. This field is available if Chatter Answers is enabled for your organization. This field is available in API version 27.0 and later.
`chatterAnswersForgotPasswordPage`	string	The name of the Visualforce page to be displayed when a user clicks the link to retrieve a forgotten password. This field is available if Chatter Answers is

Field	Field Type	Description
		enabled for your organization. This field is available in API version 27.0 and later.
`chatterAnswersHelpPage`	string	The name of the Visualforce page to be displayed when the user clicks the help link. This field is available if Chatter Answers is enabled for your organization. This field is available in API version 27.0 and later.
`chatterAnswersLoginPage`	string	The name of the Visualforce page to be displayed to allow users to log in to the portal. This field is available if Chatter Answers is enabled for your organization. This field is available in API version 27.0 and later.
`chatterAnswersRegistrationPage`	string	The name of the Visualforce page to be displayed to allow users to register themselves and access the portal. This field is available in API version 27.0 and later.
`clickjackProtectionLevel`	SiteClickjackProtectionLevel (enumeration of type string)	Required. Sets the clickjack protection level. The options are: • `AllowAllFraming` — Allow framing by any page (no protection) • `External` — Allow framing of site or Experience Cloud site pages on external domains (good protection) • `SameOriginOnly` — Allow framing by the same origin only (recommended) • `NoFraming` — Don't allow framing by any page (most protection) This field is available in API version 30.0 and later.
`contentSniffingProtection`	boolean	Required. Determines whether the browser is prevented from inferring the MIME type from the document content. If enabled, it also prevents the browser from executing some malicious files (JavaScript, Stylesheet) as dynamic content. This field is available in API version 41.0 and later.
`cspUpgradeInsecureRequests`	boolean	This field is removed in API version 52.0 and later. In API version 51.0 and earlier, the value in the field is ignored.
`customWebAddresses`	SiteWebAddress[]	The root custom URLs associated with the site. Saving or deploying a CustomSite replaces all root custom URLs in the site with the root custom URLs in this list. Custom URLs that use a non-root path prefix are not included in this list and are not affected when saving or deploying a CustomSite. This field is available in API version 21.0 and later.

Field	Field Type	Description
`description`	string	The site description.
`enableAuraRequests`	boolean	Determines whether guest users can view features available only in Lightning (`true`). If set to `false`, Lightning features don't load. This field is available in API version 46.0 and later.
`favoriteIcon`	string	The name of the file to be used for the icon that appears in the browser's address field when visiting the site. Sets the favorite icon for the entire site.
`fileNotFoundPage`	string	The name of the Visualforce page to be displayed when the guest user tries to access a non-existent page.
`forgotPasswordPage`	string	The name of the Visualforce page to be displayed when a user clicks the Forgot Password link on the site's login page. This field is only applicable for Experience Cloud sites.
`genericErrorPage`	string	The name of the Visualforce page to be displayed for errors not otherwise specified.
`guestProfile`	string	Read only. The name of the profile associated with the guest user.
`inMaintenancePage`	string	The name of the Visualforce page to be displayed when the site is down for maintenance.
`inactiveIndexPage`	string	The name of the Visualforce page set as the inactive site home page.
`indexPage`	string	Required. The name of the Visualforce page set as the active site home page.
`masterLabel`	string	Required. The name of the site label in the Salesforce user interface.
`myProfilePage`	string	The name of the Visualforce page to be displayed as the site user's profile page, where users can update their contact information. This field is available in API version 20.0 and later.
`portal`	string	The name of the portal associated with this site for login access.
`redirectToCustomDomain`	boolean	Indicates whether requests for this site's system-managed URLs are redirected to the HTTPS custom domain serving this site (`true`) or not (`false`). System-managed site URLs end in `*.my.salesforce-sites.com` or `*.my.site.com`. In Experience Cloud sites, the

Field	Field Type	Description
		default is `false`. In Salesforce Sites, the default is `true`. If multiple custom domains serve this site and this field is set to `true`, requests are routed to the site's primary custom URL only if it's an HTTPS custom domain. Otherwise, requests are redirected to the first HTTPS custom domain associated with this site, in alphanumeric order. If no HTTPS custom domain serves this site, this option has no effect. This field is available in API version 52.0 and later.
`referrerPolicyOriginWhenCrossOrigin`	boolean	Required. Determines whether the referrer header shows only Salesforce.com rather than the entire URL when loading a page. This feature eliminates the potential for a referrer header to reveal sensitive information that could be present in a full URL, such as an org ID. This field is available in API version 41.0 and later.
`requireHttps`	boolean	This field is removed in API version 52.0 and later. In API version 51.0 and earlier, the value in the field is ignored.
`requireInsecurePortalAccess`	boolean	Determines whether to override your organization's security settings and exclusively use HTTP when logging in to the associated portal from your site. Removed in API version 50.0 and later.
`robotsTxtPage`	string	The name of the Visualforce page to be displayed for the `robots.txt` file used by web crawlers.
`selfRegPage`	string	Visualforce page used for self-registration.
`serverIsDown`	string	The name of the static resource to be displayed from the cache server when Salesforce servers are down. The static resource must be a public zip file 1 MB or smaller and must contain a page named `maintenance.html` at the root level of the zip file. Other resources in the zip file, such as images or CSS files, can follow any directory structure. This field is available in API version 17.0 and later.
`siteAdmin`	string	The username of the site administrator.
`siteGuestRecordDefaultOwner`	string	The username of the user who owns all new records that unauthenticated guest users create. This field is available in API version 51.0 and later.

Field	Field Type	Description
`siteIframeWhiteListUrls`	SiteIframeWhiteListUrl[]	The list of external domains that you allow to frame your Salesforce site. This field is available in API 49.0 and later.
`siteRedirectMappings`	SiteRedirectMapping[]	An array of all URL redirect rules set for your site. This field is available in API version 20.0 and later.
`siteTemplate`	string	The name of the Visualforce page to be used as the site template.
`siteType`	siteType	Required. Identifies whether the site is a Visualforce (Salesforce Sites), Site.com site, or ChatterNetwork (Salesforce Sites).This field is available in API version 27.0 and later.
`subdomain`	string	Read only. The previous custom subdomain prefix for the site. For example, if your site URL is `mycompany.force.com/partners`, `mycompany` is the `subdomain`. This field is applicable and required only when the `myDomainSuffix` MyDomainSettings field is set to `MySalesforceLimited`, `CloudforceLimited`, or `DatabaseLimited`. If you enabled Salesforce Sites or Digital Experiences when the `myDomainSuffix` MyDomainSettings field was set to one of those values, this field returns this site's previous subdomain. Otherwise, this field returns a null value.
`urlPathPrefix`	string	The first part of the path on the site's URL that distinguishes this site from other sites. For example, if your site URL is *MyDomainName*`.my.salesforce-sites.com/partners`, `partners` is the `urlPathPrefix`.

SiteIframeWhiteListUrl

Represents the external domains that you allow to frame your site or experience pages.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. Because changing terms in our code can break current implementations, we maintained this metadata type's name.

Field	Field Type	Description
`url`	string	Required. The trusted domain that you allow to frame your site or Experience Cloud site pages. Accepts these formats: `example`,

Field	Field Type	Description
		`example.com`, `*example.com`, and `https://example.com`.

SiteRedirectMapping

SiteRedirectMapping represents a URL redirect rule on your Salesforce site." in Salesforce Help.

Field	Field Type	Description
`action`	SiteRedirect (enumeration of type string)	Required. The type of the redirect. Available string values are: • `Permanent` • `Temporary`
`isActive`	boolean	The status of the redirect: active or inactive.
`source`	string	Required. The URL that you want to redirect. It must be a relative URL, but can have any valid extension type, such as `.html` or `.php`.
`target`	string	Required. The new URL you want users to visit. It can be a relative URL or a fully-qualified URL with an `http://` or `https://` prefix.

SiteWebAddress

Represents the web address of a Salesforce site.

Field	Field Type	Description
`certificate`	string	Identifies the certificate associated with the custom domain. If the custom domain is set up for Salesforce to serve HTTPS, this field indicates which certificate to use.
`domainName`	string	Required. The domain of the website, in the form of `www.acme.com`.
`primary`	boolean	Required. Indicates whether this is the primary domain (`true`). If `false`, this is not the primary domain.

Declarative Metadata Sample Definition

Here is a sample XML definition of a site.

```
<?xml version="1.0" encoding="UTF-8"?>
<CustomSite xmlns="http://soap.sforce.com/2006/04/metadata">
    <active>true</active>
    <allowHomePage>true</allowHomePage>
    <allowStandardAnswersPages>true</allowStandardAnswersPages>
    <allowStandardIdeasPages>true</allowStandardIdeasPages>
    <allowStandardLookups>true</allowStandardLookups>
    <allowStandardPortalPages>true</allowStandardPortalPages>
    <allowStandardSearch>true</allowStandardSearch>
    <analyticsTrackingCode>UA-000000-2</analyticsTrackingCode>
    <authorizationRequiredPage>Unauthorized</authorizationRequiredPage>
    <bandwidthExceededPage>BandwidthExceeded</bandwidthExceededPage>
    <browserXssProtection>true</browserXssProtection>

<cachePublicVisualforcePagesInProxyServers>false</cachePublicVisualforcePagesInProxyServers>

    <changePasswordPage>ChangePassword</changePasswordPage>

<chatterAnswersForgotPasswordConfirmPage>ChatterAnswersForgotPasswordConfirm</chatterAnswersForgotPasswordConfirmPage>


<chatterAnswersForgotPasswordPage>ChatterAnswersForgotPassword</chatterAnswersForgotPasswordPage>

    <chatterAnswersHelpPage>ChatterAnswersHelp</chatterAnswersHelpPage>
    <chatterAnswersLoginPage>ChatterAnswersLogin</chatterAnswersLoginPage>

<chatterAnswersRegistrationPage>ChatterAnswersRegistration</chatterAnswersRegistrationPage>

    <clickjackProtectionLevel>SameOriginOnly</clickjackProtectionLevel>
    <contentSniffingProtection>true</contentSniffingProtection>
    <customWebAddresses>
      <domainName>www.testing123.com</domainName>
      <primary>true</primary>
    </customWebAddresses>
    <description>Partners portal for My Company</description>
    <enableAuraRequests>true</enableAuraRequests>
    <favoriteIcon>myFavIcon</favoriteIcon>
    <fileNotFoundPage>FileNotFound</fileNotFoundPage>
    <forgotPasswordPage>ForgotPassword</forgotPasswordPage>
    <genericErrorPage>Exception</genericErrorPage>
    <guestProfile>Guest</guestProfile>
    <inMaintenancePage>InMaintenance</inMaintenancePage>
    <inactiveIndexPage>Inactive</inactiveIndexPage>
    <indexPage>UnderConstruction</indexPage>
    <masterLabel>customSite</masterLabel>
    <myProfilePage>UserProfile</myProfilePage>
    <portal>Customer Portal</portal>
    <redirectToCustomDomain>true</redirectToCustomDomain>
    <referrerPolicyOriginWhenCrossOrigin>true</referrerPolicyOriginWhenCrossOrigin>
    <robotsTxtPage>RobotsTxt</robotsTxtPage>
    <selfRegPage>SelfReg</selfRegPage>
```

```
    <serverIsDown>MyServerDownResource</serverIsDown>
    <siteAdmin>admin@myco.org</siteAdmin>
    <siteGuestRecordDefaultOwner>admin@myco.org</siteGuestRecordDefaultOwner>
    <siteIframeWhiteListUrl>
      <url>example.com</url>
    </siteIframeWhiteListUrl>
    <siteTemplate>SiteTemplate</siteTemplate>
    <siteType>Siteforce</siteType>
    <subdomain>myco</subdomain>
    <urlPathPrefix>partners</urlPathPrefix>
</CustomSite>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SEE ALSO:

Portal

CustomTab

Represents a custom tab. Custom tabs let you display custom object data or other web content in Salesforce. When you add a custom tab to an app in Salesforce Classic, it appears as a tab. When you add a custom tab to an app in Lightning Experience, it appears as an item in the app's navigation bar and in the App Launcher. When a tab displays a custom object, the tab name is the same as the custom object name. For page, s-control, or URL tabs, the name is arbitrary.

For more information, see *Custom Tabs* in Salesforce Help. This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

The file suffix is `.tab`. There's one file for each tab, stored in the `tabs` folder in the corresponding package directory.

Note: Retrieving a component of this metadata type in a project makes the component appear in any Profile and PermissionSet components that are retrieved in the same package.

Version

Tabs are available in API version 10.0 and later.

Fields

This metadata type contains the following fields:

Field Name	Field Type	Description
`actionOverrides`	ActionOverride[]	A list of the action overrides that are assigned to the tab. Only one override is allowed per `formFactor` for a given tab. This field is available in API version 37.0 and later.
`auraComponent`	string	The name of the Aura component to display in this tab. Only one of these fields can have a value set: • `auraComponent` • `customObject` • `flexiPage` • `lwcComponent` • `page` • `scontrol` • `url`
`customObject`	boolean	Indicates whether this tab is for a custom object (`true`) or not (`false`). If set to `true`, the name of the tab matches the name of the custom object. Only one of these fields can have a value set: • `auraComponent` • `customObject` • `flexiPage` • `lwcComponent` • `page` • `scontrol` • `url`
`description`	string	The optional description text for the tab.
`flexiPage`	string	The name of the Lightning page to display in this tab. Only one of these fields can have a value set: • `auraComponent` • `customObject` • `flexiPage` • `lwcComponent` • `page` • `scontrol` • `url`
`frameHeight`	int	The height, in pixels of the tab frame. Required for s-control and page tabs.

Field Name	Field Type	Description
`fullName`	string	The name of the tab. The value of this field depends on the type of tab, and the API version. • For custom object tabs, the `fullName` is the developer-assigned name of the custom object (MyCustomObject__c, for example). For custom object tabs, this name must be the same as the custom object name, and `customObject` must be set to `true`. • For web tabs, the `fullName` is the developer-assigned name of the tab (MyWebTab, for example). The `fullName` can contain only underscores and alphanumeric characters. It must be unique, begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. This field is inherited from the Metadata component.
`hasSidebar`	boolean	Indicates if the tab displays the sidebar panel.
`icon`	string	The optional reference to the image document for the tab if the tab isn't using one of the standard tab styles. This field is available in API version 14.0.
`label`	string	The label of the tab, for web tabs only.
`lwcComponent`	string	The name of the Lightning web component to display in this tab. Only one of these fields can have a value set: • `auraComponent` • `customObject` • `flexiPage` • `lwcComponent` • `page` • `scontrol` • `url`
`motif`	string	Required. The tab style for the color scheme and icon for the custom tab. For example, "'Custom70: Handsaw," is the handsaw icon.
`page`	string	The name of the Visualforce page to display in this tab. Only one of these fields can have a value set: • `auraComponent` • `customObject` • `flexiPage` • `lwcComponent` • `page` • `scontrol` • `url`

Field Name	Field Type	Description
`scontrol`	string	The name of the s-control to display in this tab. Only one of these fields can have a value set: • `auraComponent` • `customObject` • `flexiPage` • `lwcComponent` • `page` • `scontrol` • `url`
`splashPageLink`	string	The custom link used as the introductory splash page when users click the tab. References a HomePageComponent.
`url`	string	The URL for the external web-page to embed in this tab. Only one of these fields can have a value set: • `auraComponent` • `customObject` • `flexiPage` • `lwcComponent` • `page` • `scontrol` • `url`
`urlEncodingKey`	Encoding (enumeration of type string)	The default encoding setting is Unicode: `UTF-8`. Change it if you're passing information to a URL that requires data in a different format. This option is available when the value `URL` is selected in the tab type.

Declarative Metadata Sample Definition

The following is the definition of a tab:

```
<?xml version="1.0" encoding="UTF-8"?>
<CustomTab xmlns="http://soap.sforce.com/2006/04/metadata">
    <description>Myriad Publishing</description>
    <frameHeight>600</frameHeight>
    <motif>Custom53: Bell</motif>
    <url>https://www.example.com</url>
    <urlEncodingKey>UTF-8</urlEncodingKey>
</CustomTab>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SEE ALSO:

CustomApplication

CustomValue

Represents the definition of a value used in a global value set or local custom picklist. Custom picklist fields can be local and unique, or can inherit their values from a global picklist (called a *global value set* in API version 38.0). This type extends the Metadata metadata type and inherits its `fullName` field.

To deactivate a global picklist value, you can invoke an `update()` call on GlobalPicklist (API version 37.0) or GlobalValueSet (API version 38.0 and later) with the value omitted, or with the value's `isActive` field set to `false`. Or, you can invoke an `update()` call directly on GlobalPicklistValue (API version 37.0) or CustomValue (API version 38.0 and later) with the `isActive` field set to `false`.

Note: If picklist values are missing from a component definition, they get deactivated when deployed. Deactivation occurs for picklist values of both standard and custom fields.

CustomValue doesn't support file-based operations and only supports CRUD-based calls. CustomValue is retrieved or deployed together with a GlobalValueSet or CustomObject component.

File Suffix and Directory Location

CustomValue components have the suffix `.customValue`. A CustomValue component is returned with either a GlobalValueSet or CustomObject component.

Version

CustomValue components are available in API version 38.0 and later. CustomValue replaces GlobalPicklistValue from API version 37.0.

Fields

Field Name	Field Type	Description
`color`	string	The color assigned to the picklist value when it's used in charts on reports and dashboards. The color is in hexadecimal format; for example, #FF6600. If a color isn't specified, it's assigned dynamically upon chart generation.
`default`	boolean	Required. Indicates whether this value is the default selection for the global picklist and the custom picklists that share its picklist value set. This field is set to *`true`* by default.
`description`	string	A picklist value's description. It's useful to include a description for a picklist value so the reason for creating it can be tracked. Limit: 255 characters.

Field Name	Field Type	Description
`isActive`	boolean	Indicates whether this value is active or inactive. The default value is *`true`*. Users can select only active values from a picklist. An API retrieve operation for global picklist values returns all active and inactive values in the picklist. But retrieving the values of a non-global, unrestricted picklist returns only the active values.
`label`	string	The value's display label. If you don't specify the label when creating a value it defaults to the API name. Available in API version 39.0 and later.

StandardValue

This metadata type defines a value in a value set for a standard picklist and specifies whether this value is the default value. This type extends the CustomValue metadata type and inherits all its fields.

When you deploy changes to standard picklist fields, picklist values are added as needed.

Field Name	Field Type	Description
`allowEmail`	boolean	Indicates whether this value lets users email a quote PDF (`true`), or not (`false`). This field is only relevant for the `Status` field in quotes. This field is available in API version 18.0 and later.
`closed`	boolean	Indicates whether this value is associated with a closed status (`true`), or not (`false`). This field is only relevant for the standard `Status` field in cases and tasks. This field is available in API version 16.0 and up to version 36.0. In version 37.0, this field is in GlobalPicklistValue.
`converted`	boolean	Indicates whether this value is associated with a converted status (`true`), or not (`false`). This field is relevant for only the standard `Lead Status` field in leads. Your organization can set its own guidelines for determining when a lead is qualified, but typically, you want to convert a lead as soon as it becomes a real opportunity that you want to forecast. For more information, see Convert Qualified Leads in Salesforce Help. This field is available in API version 16.0 and later.
`cssExposed`	boolean	Indicates whether this value is available in your Self-Service Portal (`true`), or not (`false`). This field is only relevant for the standard `Case Reason` field in cases. Self-Service provides an online support channel for your customers - allowing them to resolve their inquiries without contacting a customer service representative. For more information about Self-Service, see Setting Up Your Self-Service Portal in Salesforce Help. Note: Starting with Spring '12, the Self-Service portal isn't available for new Salesforce orgs. Existing orgs continue to have access to the Self-Service portal. This field is available in API version 16.0 and later.

Field Name	Field Type	Description
`forecastCategory`	ForecastCategories (enumeration of type string)	Indicates whether this value is associated with a forecast category (`true`), or not (`false`). This field is only relevant for the standard `Stage` field in opportunities. • Omitted • Pipeline • BestCase • Forecast • Closed This field is available in API version 16.0 and later.
`highPriority`	boolean	Indicates whether this value is a high priority item (`true`), or not (`false`). This field is only relevant for the standard `Priority` field in tasks. For more information about tasks, see Start Using Tasks in Salesforce Help. This field is available in API version 16.0 and later.
`probability`	int	Indicates whether this value is a probability percentage (`true`), or not (`false`). This field is only relevant for the standard `Stage` field in opportunities. This field is available in API version 16.0 and later.
`reverseRole`	string	A picklist value corresponding to a reverse role name for a partner. If the role is subcontractor, then the reverse role might be general contractor. Assigning a partner role to an account in Salesforce creates a reverse partner relationship so that both accounts list the other as a partner. This field is only relevant for partner roles. For more information, see Partner Fields in Salesforce Help. This field is available in API version 18.0 and later.
`reviewed`	boolean	Indicates whether this value is associated with a reviewed status (`true`), or not (`false`). This field is only relevant for the standard `Status` field in solutions. For more information about opportunities, see Creating Solutions in Salesforce Help. This field is available in API version 16.0 and later.
`won`	boolean	Indicates whether this value is associated with a closed or won status (`true`), or not (`false`). This field is only relevant for the standard `Stage` field in opportunities. This field is available in API version 16.0 and later.

Declarative Metadata Sample Definition

For an example of CustomValue components within a GlobalValueSet component that's referenced by a `package.xml`, see GlobalValueSet.

Dashboard

Represents a dashboard. Dashboards are visual representations of data that allow you to see key metrics and performance at a glance.

This type extends the Metadata metadata type and inherits its `fullName` field. For more information, see "Edit Dashboards in Accessibility Mode in Salesforce Classic" in the Salesforce online help.

Declarative Metadata File Suffix and Directory Location

Dashboards are stored in the `dashboards` directory of the corresponding package directory. The file name matches the dashboard title and the extension is `.dashboard`.

Retrieving Dashboards

You can't use the wildcard (*) symbol with dashboards in `package.xml`. To retrieve the list of dashboards for populating `package.xml` with explicit names, call `listMetadata()` and pass in `DashboardFolder` as the type. Note that DashboardFolder is not returned as a type in `describeMetadata()`. Dashboard is returned from `describeMetadata()` with an associated attribute of `inFolder` set to true. If that attribute is set to true, you can construct the type by using the component name with the word Folder, such as DashboardFolder.

The following example shows folders in `package.xml`. The names used in `package.xml` must be developer names, not dashboard titles.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>MyDBFolder/MyDBName</members>
        <name>Dashboard</name>
    </types>
    <types>
        <members>MyDocumentFolder/MyDocumentName</members>
        <name>Document</name>
    </types>
    <types>
        <members>unfiled$public/MarketingProductInquiryResponse</members>
        <members>unfiled$public/SalesNewCustomerEmail</members>
        <name>EmailTemplate</name>
    </types>
    <types>
        <members>MyReportFolder/MyReportName</members>
        <name>Report</name>
    </types>
    <version>64.0</version>
</Package>
```

Version

Dashboard components are available in API version 14.0 and later.

Fields

Field	Field Type	Description
backgroundEndColor	string	Required. A dashboard can have a gradient color change on its charts. This field defines the second color for the gradient and backgroundStartColor defines the first color. If you prefer your background to be all one color or do not want a gradient color change, select the same color for this field and backgroundStartColor. The color is in hexadecimal format; for example #FF6600.
backgroundFadeDirection	ChartBackgroundDirection (enumeration of type string)	Required. The direction of the gradient color change, defined by the backgroundStartColor and backgroundEndColor fields. The valid values are: • Diagonal • LeftToRight • TopToBottom
backgroundStartColor	string	Required. The starting color for the gradient color change on the dashboard's charts. See backgroundEndColor for more information. The color is in hexadecimal format; for example #FF6600.
chartTheme	ChartTheme (enumeration of type string)	Determines the default theme for all dashboard charts. Replaces dashboardChartTheme for API v42.0 and later. • light—Light-colored theme. • dark—Dark-colored theme. This field is available in API version 42.0 and later.
colorPalette	ChartColorPalettes (enumeration of type string)	Determines the default palette for all dashboard charts. Replaces dashboardColorPalette for API v42.0 and later. • accessible • bluegrass • colorSafe • Default • dusk • earth • fire • gray • heat • justice • nightfall • pond • sunrise

Field	Field Type	Description
		• `tropic` • `unity` • `water` • `watermelon` This field is available in API version 42.0 and later.
`dashboardChartTheme`	ChartTheme (enumeration of type string)	Determines the default theme for all dashboard charts. • `light`—Light-colored theme. • `dark`—Dark-colored theme. This field is available to maintain backward compatibility with versions prior to API version 42.0.
`dashboardColorPalette`	ChartColorPalettes (enumeration of type string)	Determines the default palette for all dashboard charts. • `accessible` • `bluegrass` • `colorSafe` • `Default` • `dusk` • `earth` • `fire` • `gray` • `heat` • `justice` • `nightfall` • `pond` • `sunrise` • `tropic` • `unity` • `water` • `watermelon` This field is available to maintain backward compatibility with versions prior to API version 42.0.
`dashboardFilters`	DashboardFilter[]	The list of filters in a dashboard. This field is available in API version 23.0 and later.
`dashboardGridLayout`	DashboardGridLayout	Lists the included DashboardGridComponent objects, specifies the number of dashboard columns, and sets each dashboard row's height in pixels. This field is available in API version 35.0 and later.

Field	Field Type	Description
`dashboardType`	DashboardType (enumeration of type string)	Determines the way visibility settings are set for a dashboard. The valid values are: • `SpecifiedUser`—All users see data at the access level of one specific running user, specified in the `runningUser` field, regardless of their own security settings. • `LoggedInUser`—Each logged-in user sees data according to his or her own access level. • `MyTeamUser`—Managers can choose to view the dashboard from the point of view of their subordinates in the role hierarchy. This value is available in API version 20.0 and later. This field is available in API version 19.0 and later.
`description`	string	Description for the dashboard. Maximum of 255 characters.
`folderName`	string	Name of the folder that houses the dashboard. This field is available in API version 35.0 and later.
`fullName`	string	Inherited from Metadata, this field is defined in the WSDL for this metadata type. It must be specified when creating, updating, or deleting. See `createMetadata()` to see an example of this field specified for a call. This field specifies the folder and dashboard title; for example `folderSales/California`.
`isGridLayout`	boolean	Specifies whether a dashboard uses the Lightning Experience layout (`true`) or not (`false`). Lightning Experience allows dashboards with more than three columns with components that span multiple columns and multiple rows in size. This field is available in API version 35.0 and later.
`dashboardResultRefreshedDate`	string	Required. Date that the dashboard was last refreshed.
`dashboardResultRunningUser`	string	Required. User currently accessing the dashboard.
`leftSection`	DashboardComponentSection	Required. The left section or column of the dashboard.
`middleSection`	DashboardComponentSection	The middle section or column of the dashboard.
`numSubscriptions`	int	Number of subscriptions reported on the dashboard. This field is available in API version 42.0 and later.
`rightSection`	DashboardComponentSection	Required. The right section or column of the dashboard.

Field	Field Type	Description
runningUser	string	The username of the user whose role and sharing settings are used to determine the data shown in the dashboard. When you deploy a dashboard and the value in this field is not defined or does not correspond to a valid user, the field is populated with the username of the user performing the deployment. Regardless of their security settings, all users viewing a dashboard see exactly the same data, because dashboards are always run using the security settings of a particular user. Tip: To avoid inappropriate exposure of sensitive data, save the dashboard to a folder that is visible only to appropriate users.
textColor	string	Required. Color of the text on each chart in the dashboard. The color is in hexadecimal format; for example #FF6600.
title	string	Required. The dashboard title.
titleColor	string	Required. Color of the titles on each dashboard component. The color is in hexadecimal format; for example #FF6600.
titleSize	int	Required. Size of characters in title text. For example, a value of 12 indicates 12pt text.

DashboardFilter

DashboardFilter represents a filter in a dashboard.

Field	Field Type	Description
dashboardFilterOptions	DashboardFilterOption[]	The list of items you can select in the **Filter Options** section of the Add Filter dialog.
name	string	Required. The filter label.

DashboardFilterOption

DashboardFilterOption represents a filter option in a dashboard.

Field	Field Type	Description
operator	DashboardFilterOperation (enumeration of type string)	Required. Represents the filter operation for this filter item. Valid values are: • equals • notEqual • lessThan

Field	Field Type	Description
		• `greaterThan` • `lessOrEqual` • `greaterOrEqual` • `contains` • `notContain` • `startsWith` • `includes` • `excludes` • `between` Note: The "between" operator takes two operands (for example, "between MinimumValue, MaximumValue"). Note also that the minimum value is inclusive, while the maximum value is exclusive. All other dashboard filter operations take a single operand only. This field is available in API version 24.0 and later. With API version 23.0, valid values are enumerated in CustomField.
`values`	string[]	Required. One or more values in the **Filter Options** area of the Add Filter dialog. This field is available in API version 24.0 and later.

DashboardGridLayout

Lightning Experience features dashboards with more than three columns and components that span multiple columns and multiple rows in size. DashboardGridLayout lists the included dashboard components, specifies the number of dashboard columns, and sets each dashboard row's height in pixels.

Field	Field Type	Description
`dashboardGridComponents`	DashboardGridComponent[]	List of DashboardGridComponent objects in the dashboard.
`numberOfColumns`	int	Required. Total number of columns in the dashboard.
`rowHeight`	int	Required. Height of each row in pixels.

DashboardGridComponent

Lightning Experience features dashboards with more than three columns and components that span multiple columns and multiple rows in size. DashboardGridComponent specifies location and size of a given dashboard component.

Field	Field Type	Description
`colSpan`	int	Required. The width of the dashboard component in columns. For example, if `colSpan` is 5, then the dashboard component spans five columns.
`columnIndex`	int	Required. The left-most column that is occupied by the dashboard component.
`dashboardComponent`	DashboardComponent	Required. The dashboard component that is being sized and placed.
`rowIndex`	int	Required. The top-most row that is occupied by the dashboard component.
`rowSpan`	int	Required. The height of the dashboard component in rows.

DashboardComponent

A dashboard consists of a group of different components or elements that display data. Each component can use a custom report or a custom s-control as their data source to display corporate metrics or key performance indicators. You can create several dashboard components and display them all in one dashboard aligned in up to three columns.

Field	Field Type	Description
`chartAxisRange`	ChartRangeType (enumeration of type string)	A manual or automatic axis range for bar or line charts. The valid values are: • `auto` • `manual`
`chartAxisRangeMax`	double	The maximum axis range to be displayed. This only applies to bar and line charts in which the `manual` axis range is selected for the `chartAxisRange` field.
`chartAxisRangeMin`	double	The minimum axis range to be displayed. This only applies to bar and line charts in which the `manual` axis range is selected for the `chartAxisRange` field.
`chartSummary`	ChartSummary	Specifies the summary field for the chart data. Required if `isAutoSelectFromReport` is set to `false`. This field is available in API version 25.0 and later.
`componentType`	DashboardComponentType (enumeration of type string)	Required. Dashboard component type. The valid values are: • `Bar` • `BarGrouped` • `BarStacked` • `BarStacked100`

Field	Field Type	Description
		• Column • ColumnGrouped • ColumnLine • ColumnLineGrouped • ColumnLineStacked • ColumnLineStacked100 • ColumnStacked • ColumnStacked100 • Donut • FlexTable • Funnel • Gauge • Image • Line • lineCumulative • LineGrouped • lineGroupedCumulative • Metric • Pie • PulseMetric • RichText • Scatter • ScatterGrouped • Scontrol • Table • VisualforcePage
dashboardComponentContents	DashboardComponentContent on page 771	A list of dashboard component contents. This field is available in API version 58.0 and later.
dashboardDynamicValues	DashboardDynamicValue on page 776[]	A list of dashboard dynamic values. This field is available in API version 36.0 and later.
dashboardFilterColumns	DashboardFilterColumn on page 776[]	A list of dashboard filter columns. Each report-based component must have a dashboard filter column that defines the column that the filter applies to. This field is available in API version 23.0 and later.
dashboardTableColumn	DashboardTableColumn[]	Represents a list of columns on a customized dashboard table component.

Field	Field Type	Description
`displayUnits`	ChartUnits (enumeration of type string)	Chart Units. The valid values are: • `Auto` • `Integer` • `Hundreds` • `Thousands` • `Millions` • `Billions` • `Trillions`
`drillDownUrl`	string	For charts, specifies a URL that users go to when they click the dashboard component. Use this option to send users to another dashboard, report, record detail page, or other system that uses a Web interface. This field overrides the `drillEnabled` and `drillToDetailEnabled` fields.
`drillEnabled`	boolean	Specifies whether to take users to the full or filtered source report when they click the dashboard component. Set to `false` to drill to the full source report; set to `true` to drill to the source report filtered by what they clicked. If set to `true`, users can click individual groups, axis values, or legend entries. This overrides the `drillToDetailEnabled` field. This field is available in API version 17.0 and later.
`drillToDetailEnabled`	boolean	When enabled, users are taken to the record detail page when they click a record name, record owner, or feed post in a table or chart. When set to `true` users can click axis and legend values, chart elements, and table entries. The `drillDownUrl` and `drillEnabled` fields override this field. This field is available in API version 20.0 and later.
`enableHover`	boolean	Specifies whether to display values, labels, and percentages when hovering over charts. Hover details depend on chart type. Percentages apply to pie, donut, and funnel charts only. This field is available in API version 17.0 and later.
`expandOthers`	boolean	Specifies whether to combine all groups less than or equal to 3% of the total into a single 'Others' wedge or segment. This only applies to pie, donut, and funnel charts. Set to `true` to show all values individually on the chart; set to `false` to combine small groups into 'Others.' This field is available in API version 17.0 and later.

Field	Field Type	Description
`flexComponentProperties`	DashboardFlexTableComponentProperties	Defines metadata for Lightning Experience table columns and sorting. This field is available in API version 41.0 and later.
`footer`	string	Footer displayed at the bottom of the dashboard component. Maximum of 255 characters.
`gaugeMax`	double	The maximum value on a gauge. A gauge is used to see how far you are from reaching a goal. It looks like a speedometer in a car.
`gaugeMin`	double	The minimum value on a gauge.
`groupingColumn`	string	Specifies the field by which to group data. This data is displayed on the X-axis for vertical column charts and on the Y-axis for horizontal bar charts. This field is available in API version 25.0 and later.
`GroupingSortProperties`	DashboardComponentGroupingSortProperties	This field captures sort properties of the dashboard component. If the component has one or more groupings, sort information is stored here; otherwise, it is stored in the `sortBy` field. This field is available in API version 46.0 and later.
`header`	string	Header displayed at the top of the dashboard component. Maximum of 80 characters.
`indicatorBreakpoint1`	double	The value that separates the `indicatorLowColor` from the `indicatorMiddleColor` on the dashboard.
`indicatorBreakpoint2`	double	The value that separates the `indicatorMiddleColor` from the `indicatorHighColor` on the dashboard.
`indicatorHighColor`	string	The color representing a high number range on the gauge.
`indicatorLowColor`	string	The color representing a low number range on the gauge.
`indicatorMiddleColor`	string	The color representing a medium number range on the gauge.
`legendPosition`	ChartLegendPosition (enumeration of type string)	The location of the legend with respect to the chart. The valid values are: • `Bottom` • `OnChart` • `Right`
`maxValuesDisplayed`	int	The maximum number of elements to include in the top-level grouping of the horizontal axis of a horizontal

Field	Field Type	Description
		chart, vertical axis of a vertical chart, or selected axis of a stacked bar chart. For example, if you want to list only your top five salespeople, create an opportunity report that lists total opportunity amounts by owner and enter `5` in this field.
`metricLabel`	string	Descriptive label for the metric. This is relevant if `metric` is the value of the `componentType` field.
`page`	string	Visualforce page associated with the component.
`pageHeightInPixels`	int	Display height of the Visualforce page in pixels.
`report`	string	Name of the report associated with the component.
`scontrol`	string	S-control associated with component if `scontrol` is the value of the `componentType` field. For more information, see "Defining Custom S-Controls" in the Salesforce online help.
`scontrolHeightInPixels`	int	Display height of the s-control in pixels.
`showPercentage`	boolean	Indicates if percentages are displayed for regions of gauges and wedges and segments of pie, donut, and funnel charts (`true`), or not (`false`).
`showPicturesOnCharts`	boolean	Display Chatter photos for up to 20 records in a horizontal bar chart component whose source report is grouped by a user or group name field. If there are more than 20 records with photos, record names are shown instead of photos. Set `Grouping Display` to *`None`* to show photos. Set the `Drill Down to` option to *`Record Detail Page`* to take users directly to user profile or group pages when they click photos. Chatter must be enabled for photos to be displayed. Depending on your organization's setup, you may not see photos on tables and charts.
`showPicturesOnTables`	boolean	Display Chatter photos for up to 20 records in a horizontal bar chart component whose source report is grouped by a user or group name field. If there are more than 20 records with photos, record names are shown instead of photos. Set `Grouping Display` to *`None`* to show photos. Set the `Drill Down to` option to *`Record Detail Page`* to take users directly to user profile or group pages when they click photos. Chatter must be enabled for photos to be displayed. Depending on your organization's setup, you may not see photos on tables and charts.

Field	Field Type	Description
showTotal	boolean	Indicates if the total of all wedges is displayed for gauges and donut charts (true), or not (false).
showValues	boolean	Indicates if the values of individual records or groups are displayed for charts (true), or not (false).
sortBy	DashboardComponentFilter (enumeration of type string)	The sort option for the dashboard component.
sortLegendValues	boolean	Specifies whether to sort the legend values for the dashboard component.
title	string	The title of the dashboard component. Maximum of 40 characters.
useReportChart	boolean	Specifies whether to use the chart defined in the source report on this dashboard component. The chart settings in the source report determine how the chart displays in the dashboard, and any chart settings you define for the dashboard are overridden. If you defined a combination chart in the source report, use this option to use that combination chart on this dashboard.

DashboardComponentContent

dashboardComponentContent represents the content of a dashboard's components.

Field	Field Type	Description
additionalInfo	string	Any additional metadata the user wants to include for the component contents.
altText	string	The component's alternative text.
fileName	string	The name of the component file.
fit	Fit (enumeration of type string)	The image alignment type. Valid values are: • FitHeight • FitWidth • Original • Stretch • Tile
horizontalAlignment	HorizontalAlignment (enumeration of type string)	The horizontal alignment type. Valid values are: • Left • Center • Right

Field	Field Type	Description
`richTextContent`	string	The rich text content for the component..
`tooltip`	string	The dashboard component's tooltip.
`verticalAlignment`	VerticalAlignment (enumeration of type string)	The vertical alignment type. Valid values are: • `Bottom` • `Center` • `Top`

DashboardDynamicValue

DashboardDynamicValue represents a dynamic value in a dashboard.

Field	Field Type	Description
`additionalInfo`	string	Any additional metadata the user wants to include for the dynamic value.
`fieldName`	string	Required. The name of the field for the dynamic value.
`isDynamicUser`	boolean	Indicates whether the value should be retrieved as the user running the dashboard (`true`) or not (`false`).

DashboardFilterColumn

DashboardFilterColumn represents a filter column in a dashboard.

Field	Field Type	Description
`column`	string	Required. The report column code for the filter.

DashboardTableColumn

DashboardTableColumn represents a column in a customized table component in a dashboard.

Field	Field Type	Description
`aggregateType`	ReportSummaryType[] (enumeration of type string)	Specifies the aggregation type for the table column.
`column`	string	Required. The label of the column to use in the table.
`showTotal`	boolean	Displays the totals for each summarizable column in the dashboard table. This field is available in API version 19.0 and later.

Field	Field Type	Description
`sortBy`	DashboardComponentSection (enumeration of type string)	The sort option for the dashboard table component. Sort on just one column per table.

DashboardFlexTableComponentProperties

DashboardFlexTableComponentProperties represents a column in a customized table component in a dashboard.

Field	Field Type	Description
`flexTableColumn`	DashboardComponentColumn	Represents a column in a Lightning Experience table component. This field is available in API version 41.0 and later.
`flexTableSortInfo`	DashboardComponentSortInfo	Represents sorting column and order in a Lightning Experience table component. This field is available in API version 41.0 and later.
`hideChatterPhotos`	boolean	If `true`, hides any photos from Chatter feeds. This field is available in API version 41.0 and later.
`decimalPrecision`	integer	For columns with numeric values, indicates the number of significant digits.

DashboardComponentGroupingSortProperties

DashboardComponentGroupingSortProperties is composed of multiple elements of the type DashboardComponentGroupingSort.

Field	Field Type	Description
`groupingSorts`	DashboardComponentGroupingSort	This field stores sort information for a dashboard at each grouping level of granularity. This field is available in API version 46.0 and later.

DashboardComponentGroupingSort

DashboardComponentGroupingSort specifies properties for sorting on a dashboard component group.

Field	Field Type	Description
`groupingLevel`	String	Grouping at which this sort configuration is applied.
`inheritedReportGroupingSort`	String	`true` if the sort order is picked up from an underlying report for this grouping level.
`sortColumn`	String	If grouping is sorted by an aggregate, this value is the aggregate value (such as `sortColumn`). If the grouping is sorted by its own value, this field is null.

Field	Field Type	Description
`sortOrder`	String	`Ascending` or `Descending` to reflect the sort order.

DashboardComponentColumn

DashboardComponentColumn represents a component column in a dashboard. Available in API version 41.0 and later.

Field	Field Type	Description
`breakPoint1`	double	The value that separates the `lowRangeColor` from the `midRangeColor` on the dashboard.
`breakPoint2`	double	The value that separates the `midRangeColor` from the `highRangeColor` on the dashboard.
`breakPointOrder`	double	Conditional highlighting can be applied to multiple columns. This field stores the order of conditional highlights.
`highRangeColor`	int	The color representing a high number range on the column.
`lowRangeColor`	int	The color representing a low number range on the column.
`midRangeColor`	int	The color representing a mid number range on the column.
`reportColumn`	string	Required. The report column code for the filter.
`showTotal`	boolean	If `true`, the column total is displayed.
`type`	DashboardComponentColumnType (enumeration of type string)	Represents the type of Lightning Experience table column: • `Details` • `Aggregates` • `Grouping` This field is available in API version 41.0 and later.

DashboardComponentSortInfo

DashboardFilterColumns represents a filter column in a dashboard.

Field	Field Type	Description
`ComponentSortColumn`	string	Indicates the column on which the table is sorted. This field is available in API version 41.0 and later.
`sortOrder`	string	Indicates whether column sorting is ascending or descending. This field is available in API version 41.0 and later.

DashboardComponentSection

DashboardComponentSection represents one of the sections or columns in a dashboard.

Field	Field Type	Description
`columnSize`	DashboardComponentSize (enumeration of type string)	Required. The size of the column in the dashboard: • `Medium` • `Narrow` • `Wide`
`components`	DashboardComponent[]	The list of DashboardComponent objects in the dashboard column.

DashboardComponentFilter

DashboardComponentFilter is an enumeration of type string that lists the sort values for dashboard components. The valid values are:

Enumeration Value	Description
`RowLabelAscending`	Sorts in alphabetical order by the label.
`RowLabelDescending`	Sorts in reverse alphabetical order by the label.
`RowValueAscending`	Sorts lowest to highest by the value.
`RowValueDescending`	Sorts highest to lowest by the value.

Declarative Metadata Sample Definition — Filtered Dashboard

A sample XML definition of a filtered dashboard is shown below. Note that this example is supported in API version 24.0 and later. The file name matches the dashboard title and the extension is `.dashboard`.

```
<?xml version="1.0" encoding="UTF-8"?>
<Dashboard xmlns="http://soap.sforce.com/2006/04/metadata">
    <backgroundEndColor>#FFFFFF</backgroundEndColor>
    <backgroundFadeDirection>Diagonal</backgroundFadeDirection>
    <backgroundStartColor>#FFFFFF</backgroundStartColor>
    <dashboardFilters>
        <dashboardFilterOptions>
            <operator>equals</operator>
            <values>Media</values>
        </dashboardFilterOptions>
        <dashboardFilterOptions>
            <operator>lessThan</operator>
            <values>Working</values>
        </dashboardFilterOptions>
        <dashboardFilterOptions>
            <operator>between</operator>
            <values>ABC</values>
            <values>XYZ</values>
        </dashboardFilterOptions>
        <name>Industry</name>
    </dashboardFilters>
    <dashboardFilters>
```

```
        <dashboardFilterOptions>
            <operator>equals</operator>
            <values>Analyst,Partner</values>
        </dashboardFilterOptions>
        <dashboardFilterOptions>
            <operator>startsWith</operator>
            <values>Integrator</values>
        </dashboardFilterOptions>
        <name>Account Type</name>
    </dashboardFilters>
    <dashboardType>SpecifiedUser</dashboardType>
    <leftSection>
        <columnSize>Medium</columnSize>
        <components>
            <chartAxisRange>Auto</chartAxisRange>
            <componentType>Bar</componentType>
            <dashboardFilterColumns>
                <column>INDUSTRY</column>
            </dashboardFilterColumns>
            <dashboardFilterColumns>
                <column>TYPE</column>
            </dashboardFilterColumns>
            <displayUnits>Auto</displayUnits>
            <drillEnabled>false</drillEnabled>
            <drillToDetailEnabled>false</drillToDetailEnabled>
            <enableHover>false</enableHover>
            <expandOthers>false</expandOthers>
            <legendPosition>Bottom</legendPosition>
            <report>unfiled$public/SampleReportofAccounts</report>
            <showPercentage>false</showPercentage>
            <showPicturesOnCharts>false</showPicturesOnCharts>
            <showValues>false</showValues>
            <sortBy>RowLabelAscending</sortBy>
            <useReportChart>false</useReportChart>
        </components>
    </leftSection>
    <middleSection>
        <columnSize>Medium</columnSize>
        <components>
            <chartAxisRange>Auto</chartAxisRange>
            <componentType>Funnel</componentType>
            <dashboardFilterColumns>
                <column>ACCOUNT_INDUSTRY</column>
            </dashboardFilterColumns>
            <dashboardFilterColumns>
                <column>ACCOUNT.TYPE</column>
            </dashboardFilterColumns>
            <displayUnits>Auto</displayUnits>
            <drillEnabled>false</drillEnabled>
            <drillToDetailEnabled>false</drillToDetailEnabled>
            <enableHover>false</enableHover>
            <expandOthers>false</expandOthers>
            <legendPosition>Bottom</legendPosition>
            <report>unfiled$public/SampleReportofCases</report>
```

```
            <showPercentage>false</showPercentage>
            <showValues>true</showValues>
            <sortBy>RowLabelAscending</sortBy>
            <useReportChart>false</useReportChart>
        </components>
    </middleSection>
    <rightSection>
        <columnSize>Medium</columnSize>
        <components>
            <chartAxisRange>Auto</chartAxisRange>
            <componentType>Column</componentType>
            <dashboardFilterColumns>
                <column>INDUSTRY</column>
            </dashboardFilterColumns>
            <dashboardFilterColumns>
                <column>ACCOUNT_TYPE</column>
            </dashboardFilterColumns>
            <displayUnits>Auto</displayUnits>
            <drillEnabled>false</drillEnabled>
            <drillToDetailEnabled>false</drillToDetailEnabled>
            <enableHover>false</enableHover>
            <expandOthers>false</expandOthers>
            <legendPosition>Bottom</legendPosition>
            <report>unfiled$public/SampleReportofOpportunities</report>
            <showPercentage>false</showPercentage>
            <showValues>false</showValues>
            <sortBy>RowLabelAscending</sortBy>
            <useReportChart>false</useReportChart>
        </components>
    </rightSection>
    <runningUser>admin@TESTORGNUM</runningUser>
    <textColor>#000000</textColor>
    <title>My Dashboard</title>
    <titleColor>#000000</titleColor>
    <titleSize>12</titleSize>
</Dashboard>
```

Declarative Metadata Sample Definition — Unfiltered Dashboard

A sample XML definition of a dashboard is shown below. The file name matches the dashboard title and the extension is `.dashboard`.

```
<?xml version="1.0" encoding="UTF-8"?>
<Dashboard xmlns="http://soap.sforce.com/2006/04/metadata">
    <backgroundEndColor>#FFFFFF</backgroundEndColor>
    <backgroundFadeDirection>LeftToRight</backgroundFadeDirection>
    <backgroundStartColor>#FFFFFF</backgroundStartColor>
    <description>Dashboard with all possible chart types</description>
    <leftSection>
        <columnSize>Medium</columnSize>
        <components>
            <chartAxisRange>Auto</chartAxisRange>
            <componentType>BarStacked100</componentType>
            <displayUnits>Auto</displayUnits>
            <drillEnabled>true</drillEnabled>
```

```
        <enableHover>true</enableHover>
        <report>testFolder/sourceRep</report>
        <sortBy>RowLabelAscending</sortBy>
    </components>
    <components>
        <componentType>Table</componentType>
        <dashboardTableColumn>
            <column>CLOSE_DATE</column>
            <sortBy>RowLabelAscending</sortBy>
        </dashboardTableColumn>
        <dashboardTableColumn>
            <aggregateType>Sum</aggregateType>
            <column>AMOUNT</column>
            <showTotal>true</showTotal>
        </dashboardTableColumn>
        <dashboardTableColumn>
            <column>STAGE_NAME</column>
        </dashboardTableColumn>
        <dashboardTableColumn>
            <column>PROBABILITY</column>
            <aggregateType>Maximum</aggregateType>
        </dashboardTableColumn>
        <displayUnits>Integer</displayUnits>
        <header>Opportunities Table</header>
        <indicatorHighColor>#54C254</indicatorHighColor>
        <indicatorLowColor>#C25454</indicatorLowColor>
        <indicatorMiddleColor>#C2C254</indicatorMiddleColor>
        <maxValuesDisplayed>10</maxValuesDisplayed>
        <report>testFolder/sourceRep</report>
    </components>
    <components>
        <chartAxisRange>Auto</chartAxisRange>
        <componentType>Bar</componentType>
        <displayUnits>Auto</displayUnits>
        <drillEnabled>true</drillEnabled>
        <enableHover>true</enableHover>
        <report>testFolder/sourceRep</report>
        <sortBy>RowLabelAscending</sortBy>
    </components>
    <components>
        <chartAxisRange>Auto</chartAxisRange>
        <componentType>Column</componentType>
        <displayUnits>Auto</displayUnits>
        <drillEnabled>true</drillEnabled>
        <legendPosition>Bottom</legendPosition>
        <report>testFolder/sourceRep</report>
        <sortBy>RowLabelAscending</sortBy>
        <useReportChart>true</useReportChart>
    </components>
    <components>
        <chartAxisRange>Auto</chartAxisRange>
        <componentType>Funnel</componentType>
        <displayUnits>Auto</displayUnits>
        <drillEnabled>true</drillEnabled>
```

```
            <enableHover>true</enableHover>
            <expandOthers>true</expandOthers>
            <legendPosition>Bottom</legendPosition>
            <report>testFolder/sourceRep</report>
            <sortBy>RowLabelAscending</sortBy>
        </components>
    </leftSection>
    <middleSection>
        <columnSize>Medium</columnSize>
        <components>
            <chartAxisRange>Auto</chartAxisRange>
            <componentType>ColumnStacked100</componentType>
            <displayUnits>Auto</displayUnits>
            <drillEnabled>true</drillEnabled>
            <enableHover>true</enableHover>
            <report>testFolder/sourceRep</report>
            <sortBy>RowLabelAscending</sortBy>
        </components>
        <components>
            <chartAxisRange>Auto</chartAxisRange>
            <componentType>ColumnStacked</componentType>
            <displayUnits>Auto</displayUnits>
            <drillEnabled>true</drillEnabled>
            <enableHover>true</enableHover>
            <report>testFolder/sourceRep</report>
            <sortBy>RowLabelAscending</sortBy>
        </components>
        <components>
            <chartAxisRange>Auto</chartAxisRange>
            <componentType>ColumnStacked</componentType>
            <displayUnits>Auto</displayUnits>
            <drillEnabled>true</drillEnabled>
            <enableHover>true</enableHover>
            <report>testFolder/sourceRep</report>
            <sortBy>RowLabelAscending</sortBy>
        </components>
        <components>
            <chartAxisRange>Auto</chartAxisRange>
            <componentType>ColumnGrouped</componentType>
            <displayUnits>Auto</displayUnits>
            <drillEnabled>true</drillEnabled>
            <enableHover>true</enableHover>
            <report>testFolder/sourceRep</report>
            <sortBy>RowLabelAscending</sortBy>
        </components>
        <components>
            <chartAxisRange>Auto</chartAxisRange>
            <componentType>Column</componentType>
            <displayUnits>Auto</displayUnits>
            <drillEnabled>true</drillEnabled>
            <enableHover>true</enableHover>
            <report>testFolder/sourceRep</report>
            <sortBy>RowLabelAscending</sortBy>
        </components>
```

```
    </middleSection>
    <rightSection>
        <columnSize>Medium</columnSize>
        <components>
            <chartAxisRange>Auto</chartAxisRange>
            <componentType>Bar</componentType>
            <displayUnits>Auto</displayUnits>
            <drillEnabled>true</drillEnabled>
            <enableHover>true</enableHover>
            <report>testFolder/sourceRep</report>
            <sortBy>RowLabelAscending</sortBy>
        </components>
        <components>
            <chartAxisRange>Auto</chartAxisRange>
            <componentType>Pie</componentType>
            <displayUnits>Auto</displayUnits>
            <drillEnabled>true</drillEnabled>
            <enableHover>true</enableHover>
            <expandOthers>true</expandOthers>
            <report>testFolder/sourceRep</report>
            <sortBy>RowLabelAscending</sortBy>
        </components>
        <components>
            <chartAxisRange>Auto</chartAxisRange>
            <componentType>LineGroupedCumulative</componentType>
            <displayUnits>Auto</displayUnits>
            <drillEnabled>true</drillEnabled>
            <enableHover>true</enableHover>
            <report>testFolder/sourceRep</report>
            <sortBy>RowLabelAscending</sortBy>
        </components>
        <components>
            <chartAxisRange>Auto</chartAxisRange>
            <componentType>LineGrouped</componentType>
            <displayUnits>Auto</displayUnits>
            <drillEnabled>true</drillEnabled>
            <enableHover>true</enableHover>
            <report>testFolder/sourceRep</report>
            <sortBy>RowLabelAscending</sortBy>
        </components>
        <components>
            <chartAxisRange>Auto</chartAxisRange>
            <componentType>LineCumulative</componentType>
            <displayUnits>Auto</displayUnits>
            <drillEnabled>true</drillEnabled>
            <enableHover>true</enableHover>
            <report>testFolder/sourceRep</report>
            <sortBy>RowLabelAscending</sortBy>
        </components>
        <components>
            <chartAxisRange>Auto</chartAxisRange>
            <componentType>Donut</componentType>
            <displayUnits>Auto</displayUnits>
            <drillEnabled>true</drillEnabled>
```

```
            <enableHover>true</enableHover>
            <expandOthers>true</expandOthers>
            <report>testFolder/sourceRep</report>
            <sortBy>RowLabelAscending</sortBy>
        </components>
    </rightSection>
    <runningUser>admin@TESTORGNUM</runningUser>
    <textColor>#000000</textColor>
    <title>Db Title</title>
    <titleColor>#000000</titleColor>
    <titleSize>12</titleSize>
</Dashboard>
```

Declarative Metadata Sample Definition — Lightning Experience Dashboard with `isGridLayout` Equals `true`

A sample XML definition of a Lightning Experience dashboard with `isGridLayout` equals `true` is shown below. Note that this example is supported in API version 35.0 and later. The file name matches the dashboard title and the extension is `.dashboard`.

```
<?xml version="1.0" encoding="UTF-8"?>
<Dashboard xmlns="http://soap.sforce.com/2006/04/metadata">
    <backgroundEndColor>#FFFFFF</backgroundEndColor>
    <backgroundFadeDirection>Diagonal</backgroundFadeDirection>
    <backgroundStartColor>#FFFFFF</backgroundStartColor>
    <dashboardType>SpecifiedUser</dashboardType>
    <gridLayout>
        <dashboardGridComponents>
            <colSpan>3</colSpan>
            <columnIndex>0</columnIndex>
            <dashboardComponent>
                <autoselectColumnsFromReport>false</autoselectColumnsFromReport>
                <chartAxisRange>Auto</chartAxisRange>
                <chartSummary>
                    <axisBinding>y</axisBinding>
                    <column>RowCount</column>
                </chartSummary>
                <componentType>Donut</componentType>
                <drillEnabled>false</drillEnabled>
                <drillToDetailEnabled>false</drillToDetailEnabled>
                <enableHover>false</enableHover>
                <expandOthers>false</expandOthers>
                <groupingColumn>TITLE</groupingColumn>
                <legendPosition>Bottom</legendPosition>
                <report>unfiled$public/lead_rpt</report>
                <showPercentage>false</showPercentage>
                <showTotal>false</showTotal>
                <showValues>true</showValues>
                <sortBy>RowLabelAscending</sortBy>
                <useReportChart>false</useReportChart>
            </dashboardComponent>
            <rowIndex>0</rowIndex>
            <rowSpan>3</rowSpan>
        </dashboardGridComponents>
```

```
    <dashboardGridComponents>
        <colSpan>3</colSpan>
        <columnIndex>0</columnIndex>
        <dashboardComponent>
            <autoselectColumnsFromReport>false</autoselectColumnsFromReport>
            <chartAxisRange>Auto</chartAxisRange>
            <chartSummary>
                <axisBinding>y</axisBinding>
                <column>RowCount</column>
            </chartSummary>
            <componentType>Pie</componentType>
            <drillEnabled>false</drillEnabled>
            <drillToDetailEnabled>false</drillToDetailEnabled>
            <enableHover>false</enableHover>
            <expandOthers>false</expandOthers>
            <groupingColumn>TITLE</groupingColumn>
            <legendPosition>Bottom</legendPosition>
            <report>unfiled$public/lead_rpt</report>
            <showPercentage>false</showPercentage>
            <showValues>true</showValues>
            <sortBy>RowLabelAscending</sortBy>
            <useReportChart>false</useReportChart>
        </dashboardComponent>
        <rowIndex>3</rowIndex>
        <rowSpan>3</rowSpan>
    </dashboardGridComponents>
    <dashboardGridComponents>
        <colSpan>3</colSpan>
        <columnIndex>0</columnIndex>
        <dashboardComponent>
            <autoselectColumnsFromReport>false</autoselectColumnsFromReport>
            <chartAxisRange>Auto</chartAxisRange>
            <chartSummary>
                <axisBinding>y</axisBinding>
                <column>RowCount</column>
            </chartSummary>
            <componentType>Column</componentType>
            <drillEnabled>false</drillEnabled>
            <drillToDetailEnabled>false</drillToDetailEnabled>
            <enableHover>false</enableHover>
            <expandOthers>false</expandOthers>
            <groupingColumn>TITLE</groupingColumn>
            <legendPosition>Bottom</legendPosition>
            <report>unfiled$public/lead_rpt</report>
            <showPercentage>false</showPercentage>
            <showValues>false</showValues>
            <sortBy>RowLabelAscending</sortBy>
            <useReportChart>false</useReportChart>
        </dashboardComponent>
        <rowIndex>9</rowIndex>
        <rowSpan>3</rowSpan>
    </dashboardGridComponents>
    <numberOfColumns>9</numberOfColumns>
    <rowHeight>90</rowHeight>
```

```
    </gridLayout>
    <isGridLayout>true</isGridLayout>
    <runningUser>admin@s1.com</runningUser>
    <textColor>#000000</textColor>
    <title>sfx</title>
    <titleColor>#000000</titleColor>
    <titleSize>12</titleSize>
</Dashboard>
```

Wildcard Support in the Manifest File

This metadata type doesn't support the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SEE ALSO:

- Folder
- Report

DataCategoryGroup

Represents a data category group.

This type extends the Metadata metadata type and inherits its `fullName` field.

Warning: Using Metadata API to deploy category changes from one organization to another permanently removes categories and record categorizations that are not specified in your XML file. Salesforce recommends that you manually create data categories and record associations in an organization from Setup by entering *`Data Categories`* in the `Quick Find` box, then selecting **Data Categories** rather than deploying changes from a sandbox to a production organization. For more information, see Usage.

Data category groups are provided to:

- Classify and filter data.
- Share data among users.

Every data category group contains items or data categories that can be organized hierarchically.

The example below shows the `Geography` data category group and its data categories.

```
Geography
    Worldwide
        North America
            United States of America
            Canada
            Mexico
        Europe
        Asia
```

Note: See "Work with Data Categories" in the Salesforce online help for more information on data category groups, data categories, parent and sub categories.

File Suffix and Directory Location

The file suffix is `.datacategorygroup`. There is one file for each data category group stored in the `datacategorygroups` folder in the corresponding package directory.

Version

Data category groups are available in API version 18.0 and later.

Fields

This metadata type contains the following fields:

Field Name	Field Type	Description
`active`	boolean	Required. The status of the category group. Indicates whether this category group is active, (`true`), or not active (`false`).
`dataCategory`	DataCategory on page 788	Required. The top-level category within the data category group.
`description`	string	The description of the data category group.
`fullName`	string	Required. The unique name of the data category group. When creating a data category group, the `fullName` field and the file name (without its suffix) must match.The `fullName` can contain only underscores and alphanumeric characters. It must be unique, begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. This field is inherited from the Metadata component.
`label`	string	Required. Label that represents the object in Salesforce.
`objectUsage`	ObjectUsage on page 789	The objects that are associated with the data category group.

DataCategory

Represents an item (or data category) in the data category group. A data category can recursively contain a list of other data categories.

Field Name	Field Type	Description
`dataCategory`	DataCategory[]	A recursive list of sub data categories. For example, a list of countries within a continent. You can create up to 100 categories in a data category group and have up to 5 levels in a data category group hierarchy.
`label`	string	Required. Label for the data category throughout the Salesforce user interface.
`name`	string	Required. The developer name of the data category used as a unique identifier for API access. The name can only contain characters, letters,

Field Name	Field Type	Description
		and the underscore (_) character, must start with a letter, and cannot end with an underscore or contain two consecutive underscore characters. **Important:** The value for this field is defined once and cannot be changed later. **Warning:** If you deploy a category group that already exists in an organization, any category that is not defined in the XML file is permanently removed from your organization. For more information see Usage.

ObjectUsage

Represents the objects that can be associated with the data category group. This association allows the object to be classified and filtered using the data categories.

Field Name	Field Type	Description
`object`	string[]	A list of the object names that can be associated with the data category group. Valid values are: • `KnowledgeArticleVersion`—to associate articles. See "Modify Default Category Group Assignments for Articles" in the Salesforce online help for more information on data category groups association to articles. • `Question`—to associate questions. You can associate the `Question` object with at most one category group. **Warning:** If you deploy a category group that already exists in an organization, any object association that is not defined in the XML file is permanently removed from your organization. Ensure that your XML file specifies all the records associated with your category group in the organization. For more information see Usage.

Declarative Metadata Sample Definition

This sample is the definition of the `Geography` data category group and its data categories:

```
<?xml version="1.0" encoding="UTF-8"?>
<DataCategoryGroup xmlns="http://soap.sforce.com/2006/04/metadata">
   <label>Geography</label>
   <description>Geography structure of service center locations</description>
   <fullName>geo</fullName>

   <dataCategory> <name>WW</name> <label>Worldwide</label>
      <dataCategory> <name>AMER</name> <label>North America</label>
```

```
        <dataCategory>
            <name>USA</name>
            <label>United States of America</label>
        </dataCategory>
        <dataCategory>
            <name>CAN</name>
            <label>Canada</label>
        </dataCategory>
        <dataCategory>
            <name>MEX</name>
            <label>Mexico</label>
        </dataCategory>
    </dataCategory>
    <dataCategory> <name>EMEA</name> <label>Europe, Middle East, Africa</label>
        <dataCategory>
            <name>FR</name>
            <label>France</label>
        </dataCategory>
        <dataCategory>
            <name>SP</name>
            <label>Spain</label>
       </dataCategory>
        <dataCategory>
           <name>UK</name>
           <label>United-Kingdom</label>
        </dataCategory>
    </dataCategory>
    <dataCategory>
        <name>APAC</name>
        <label>Asia</label>
    </dataCategory>
  </dataCategory>

  <objectUsage>
    <object>KnowledgeArticleVersion </object>
  <objectUsage>
</DataCategoryGroup>
```

Usage

When you deploy a category group XML file, Metadata API checks whether the category group exists in the target organization. If the category group does not exist, it is created. If the category group already exists, then Metadata API:

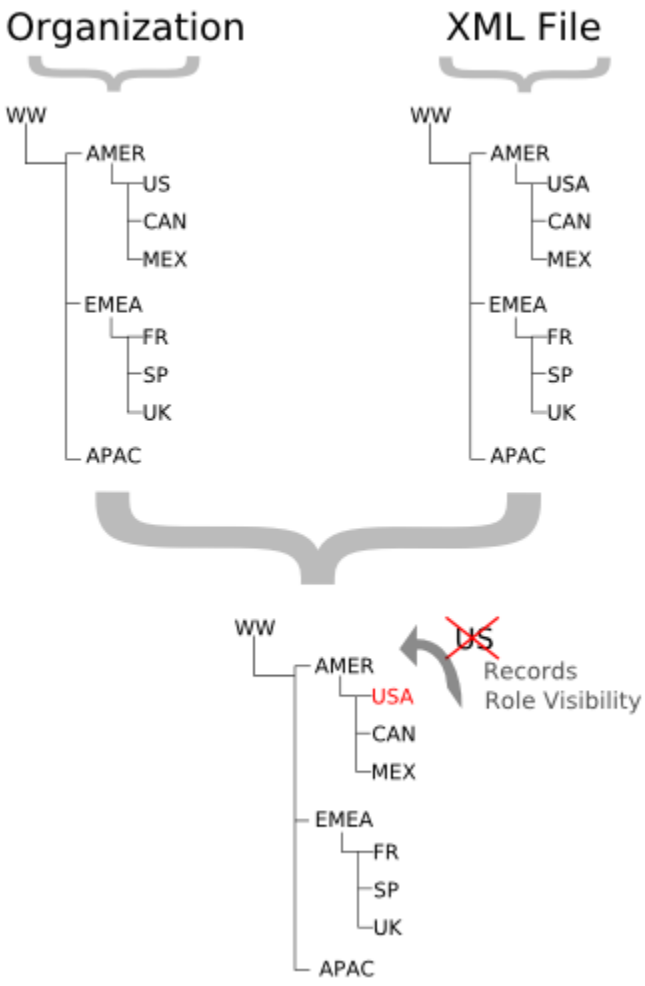
- Adds any new category or object defined in the XML file.
- Deletes any category that is not defined in the XML file. Records associated with the deleted categories are re-associated with the parent category.
- Deletes any object association that is not defined in the XML file.
- Moves any category if its hierarchical position differs from the position specified in the XML file.

  Note: When a category moves to a new parent category, users that have no visibility on the new parent category lose their visibility to the repositioned category.

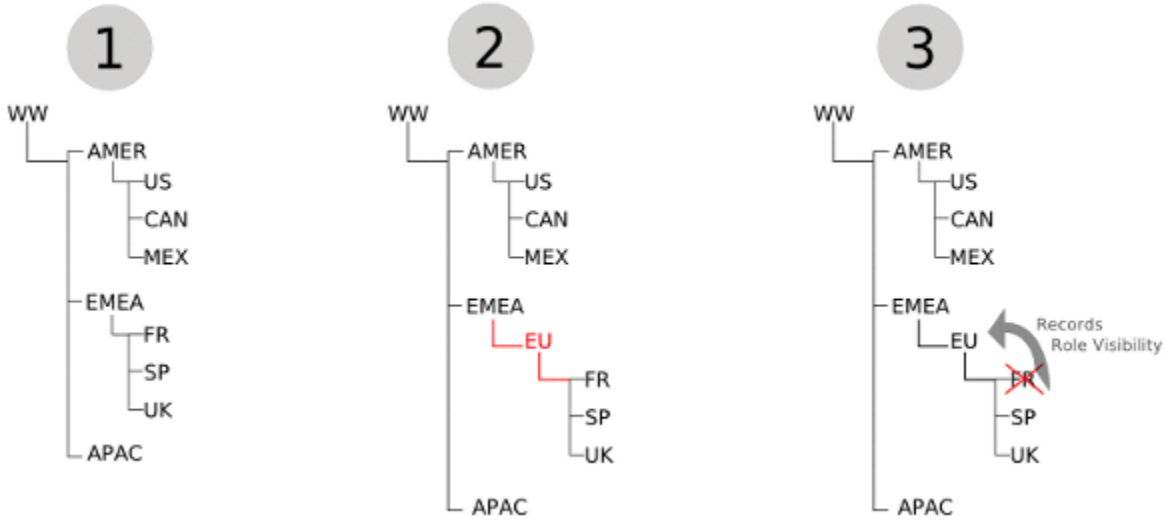
Note: For more information about category deletion, category repositioning and its impact on record categorization and visibility see "Delete a Data Category" and "Modify and Arrange Data Categories" in the Salesforce online help.

Using Metadata API to deploy category changes from one organization to another permanently removes categories and record categorizations that are not specified in your XML file. Salesforce recommends that you manually create data categories and record associations in an organization from Setup by entering *Data Categories* in the Quick Find box, then selecting **Data Categories** rather than deploying changes from a sandbox to a production organization.

The following example illustrates what happens if you deploy an XML representation of a Geography data category group hierarchy to an organization that already has this data category group defined. Note that the organization contains a US category, while the XML file includes a USA category in the same hierarchical position. The Metadata API deployment process deletes the US category from the organization and moves associations for any records from US to the parent AMER category. It also adds the USA category under AMER. Note that all records that were previously categorized with US are now associated with the AMER category.



The next example illustrates what can happen when you delete or move a category in a data category group and deploy its XML representation from a sandbox to a production organization that already has this data category group defined. Hierarchy 1 shows the initial data category group in the sandbox organization. In hierarchy 2, we add an EU category under EMEA and move FR, SP and UK below EU. In hierarchy 3, we delete FR and associate its records with its new parent, EU. Finally, we deploy the changes from the sandbox to the production organization.



Metadata API has no concept of the order of the changes made to the sandbox organization. It just deploys the changes from one organization to another. During the deployment, it first notices the deletion of the `FR` category and removes it from the production organization. Consequently, it moves associations for any records from `FR` to its parent on the production organization, `EMEA`. Metadata API then adds the `EU` category and moves `SP` and `UK` below it. Although the category group hierarchy looks the same in both organizations, record categorization in production is different from the sandbox organization. The records that were originally associated with `FR` in hierarchy 1 are associated with `EU` in the sandbox organization, but are associated with `EMEA` in the production organization.

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

DataObjectSearchIndexConf

Represents the source Data Cloud data model object (DMO) for Search Answers and holds the search index that Search Answers uses when searching DMO records.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

DataObjectSearchIndexConf components have the suffix `.dataObjectSearchIndexConf` and are stored in the `dataObjectSearchIndexConfs` folder.

Version

DataObjectSearchIndexConf components are available in API version 63.0 and later.

Special Access Rules

To access this metadata type, you must have the Customize Application user permission. The Salesforce org must have a Data Cloud license.

Fields

Field Name	Description
`application`	**Field Type** string **Description** Required. The name of the app that the Search Answers index is associated with.
`channel`	**Field Type** string **Description** The search channel that the Search Answers configuration applies to.
`masterLabel`	**Field Type** string **Description** Required. The name of the Search Answers configuration.
`nameFieldReference`	**Field Type** string **Description** Required. The name field of the DMO selected as a source for Search Answers.
`objectReference`	**Field Type** string **Description** Required. The DMO that the Search Answers configuration applies to.
`retriever`	**Field Type** string

Field Name	Description
	Description The retriever that accesses the Search Answers indexed data.
`searchIndex`	**Field Type** string **Description** Required. The name of the search index mapped to the DMO.

Declarative Metadata Sample Definition

The following is an example of a DataObjectSearchIndexConf component.

```
<?xml version="1.0" encoding="UTF-8"?>
<DataObjectSearchIndexConf xmlns="http://soap.sforce.com/2006/04/metadata">
 <application>SearchAnswers</application>
 <channel>SharedIndex</channel>
 <masterLabel>SearchAnswers</masterLabel>
 <nameFieldReference>Name__c</nameFieldReference>
 <objectReference>Account__dlm</objectReference>
 <searchIndex>searchAnswersIndex</searchIndex>
</DataObjectSearchIndexConf>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
   <types>
       <members>*</members>
       <name>DataObjectSearchIndexConf</name>
   </types>
   <version>63.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

DataWeaveResource

Represents the `DataWeaveScriptResource` class that is generated for all DataWeave scripts. DataWeave scripts can be directly invoked from Apex.

Parent Type

This type extends the MetadataWithContent metadata type and inherits its `content` and `fullName` fields.

File Suffix and Directory Location

DataWeaveResource components have the suffix `.dwl` and are stored in the `dw` folder.

Version

DataWeaveResource components are available in API version 58.0 and later.

Special Access Rules

There are no additional access requirements that are specific to this type.

Fields

Field Name	Description
`apiVersion`	**Field Type** double **Description** Required. The API version for this component.
`isGlobal`	**Field Type** boolean **Description** When set to `true`, the generated `DataWeaveScriptResource` class is global.
`isProtected`	**Field Type** boolean **Description** Not used.

Declarative Metadata Sample Definition

The following is an example of a DataWeaveResource component.

`csvToContacts.dwl`

```
%dw 2.0
input records application/csv
output application/apex
```

```
---
records map(record) -> {
 FirstName: record.first_name,
 LastName: record.last_name,
 Email: record.email
} as Object {class: "Contact"}
```

`csvToContacts.dwl-meta.xml`

```
<?xml version="1.0" encoding="UTF-8"?>
<DataWeaveResource xmlns="http://soap.sforce.com/2006/04/metadata">
    <apiVersion>58.0</apiVersion>
    <isGlobal>true</isGlobal>
</DataWeaveResource>
```

The following is an example `package.xml` that references the csvToContacts definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package
      xmlns="http://soap.sforce.com/2006/04/metadata">
      <types>
            <members>csvToContacts</members>
            <name>DataWeaveResource</name>
      </types>
      <version>58.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

DecisionTable

Represents the information about a decision table.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

DecisionTable components have the suffix `.decisionTable` and are stored in the `decisionTables` folder.

Version

DecisionTable components are available in API version 51.0 and later.

Special Access Rules

To use this metadata type, your Salesforce org must have the Loyalty Management or the Rebate Management license.

Fields

Field Name	Field Type	Description
`collectOperator`	DecisionTable CollectOperator (enumeration of type string)	Specifies the operator that's used when the result is filtered by the Collect operator. Valid values are: • `Count` • `Maximum` • `Minimum` • `None` • `Sum`
`conditionCriteria`	string	Logic that's used to decide how the input fields are processed.
`conditionType`	DecisionTable ConditionType (enumeration of type string)	Condition logic that's used for input fields. Valid values are: • `All` • `Any` • `Custom`
`dataSourceType`	DecisionTable DataSourceType (enumeration of type string)	Required. Specifies the type of data source that's used to create a decision table. Valid values are: • `CsvUpload` • `MultipleSobjects` • `SingleSobject`
`decisionTableParameters`	DecisionTableParameter[]	Required. Parameters that you specify in a decision table.
`description`	string	Description of the decision table.
`doesConsiderNullValue`	boolean	Indicates whether a column that has a null value is considered for lookup (`true`) or not (`false`). The default value is false.
`executionType`	DecisionTable executionType (enumeration of type string)	Indicates the backing storage for the Decision Table. Valid values are: • `Hbase` • `Soql`

Field Name	Field Type	Description
`filterResultBy`	DecisionTable HitPolicy (enumeration of type string)	Specifies how the results of a decision table are filtered if a set of inputs returns multiple matching outputs. Valid values are: • `AnyValue` • `FirstMatch` • `OutputOrder`
`hasIncrementalSyncFailed`	boolean	Indicates if the last incremental refresh failed.
`isIncrementalSyncEnabled`	boolean	Indicates if incremental refresh is enabled for the Decision Table.
`lastSyncDate`	string	Required. Latest date on which the decision table was refreshed.
`refreshFailureReason`	string	Reason why the refresh of the decision table data failed.
`refreshStatus`	DecisionTable RefreshStatus (enumeration of type string)	Specifies the refresh status of the cached data in the decision table. Valid values are: • `Completed` • `Failed` • `InProgress` • `Initiated`
`setupName`	string	Required. Name of the decision table, which appears in Salesforce Setup.
`sourceObject`	string	Required. Object that contains the rules based on which the decision table must provide outcomes.
`status`	DecisionTableStatus (enumeration of type string)	Required. Status of the decision table. Valid values are: • `Draft` • `Inactive` • `Active` • `ActivationInProgress`
`type`	DecisionTableType Operator (enumeration of type string)	Required. Stores the type of decision table. Valid values are: • `Advanced` • `High Volume` • `Low Volume` • `Medium Volume` • `High-Scale Execution`

Field Name	Field Type	Description
usageType	ExpsSetProcessType (enumeration of type string)	Type of industry or the application within the industry that's using a decision table. Valid values are: • Bre Note: When Business Rules Engine is enabled for a Salesforce instance, the default value is 'Bre'. Other usage types may be available to you depending on your industry solution and permission sets. • GpaCalculation • PlanCostCalculation • RatingDiscovery • StudentSuccess

DecisionTableParameters

Represents an input or output field of a decision table.

Fields

Field Name	Field Type	Description
dataType	DTParameterDataType (enumeration of type string)	The data type of the field used in a decision table. Valid values are: • Boolean • Currency • Date • Number • Percent • String
fieldName	string	Required. API name of the fields that selected as an input or output for the decision table.
isGroupByField	boolean	Indicates whether an input field is used to group the business rules of the decision table.
operator	DecisionTable Operator (enumeration of type string)	Required. Operator used for the input field. Valid values are: • Contains • DoesNotExistIn • DoesNotMatch • Equals • ExistsIn

Field Name	Field Type	Description
		• GreaterOrEqual • GreaterThan • LessOrEqual • LessThan • Matches • NotEquals
sequence	integer	The sequence in which input fields are processed. This field is available in API version 52.0 and later.
sortType	DecisionTable SortType (enumeration of type string)	Sort outputs of a decision table based on the values of the input or output parameter field. This field is available in API version 56.0 and later. Valid values are: • AscNullFirst • AscNullLast • DescNullFirst • DescNullLast • None Outputs can't be sorted based on picklist and multi-select picklist fields.
usage	DecisionTable ParameterType (enumeration of type string)	Required. Usage type of a field. The usage type can be one of the following values: • INPUT • OUTPUT

Declarative Metadata Sample Definition

The following is an example of a DecisionTable component.

```
<?xml version="1.0" encoding="UTF-8"?>
<DecisionTable xmlns="http://soap.sforce.com/2006/04/metadata">
    <collectOperator>None</collectOperator>
    <conditionCriteria>1 and 2 and 3 and 4</conditionCriteria>
    <conditionType>All</conditionType>
    <dataSourceType>SingleSobject</dataSourceType>
    <decisionTableParameters>
        <fieldName>IsDeleted</fieldName>
        <operator>Equals</operator>
        <usage>INPUT</usage>
        <sequence>1</sequence>
        <isGroupByField>true</isGroupByField>
        <sortType>AscNullFirst</sortType>
        <dataType>Number</dataType>
        <fieldPath>AccountFeed.CommentsCount</fieldPath>
        <domainObject>AccountFeed</domainObject>
```

```
        <isPriorityField>false</isPriorityField>
        <decimalScale>2</decimalScale>
        <length>14</length>
        <isRequired>false</isRequired>
    </decisionTableParameters>
    <decisionTableParameters>
        <fieldName>IsActive</fieldName>
        <usage>OUTPUT</usage>
    </decisionTableParameters>
    <decisionTableParameters>
        <fieldName>LimitNumber</fieldName>
        <operator>Equals</operator>
        <usage>INPUT</usage>
        <sequence>2</sequence>
        <isGroupByField>false</isGroupByField>
    </decisionTableParameters>
    <decisionTableParameters>
        <fieldName>LimitStartDate</fieldName>
        <usage>OUTPUT</usage>
    </decisionTableParameters>
    <decisionTableParameters>
        <fieldName>GivenBadgeCount</fieldName>
        <operator>Equals</operator>
        <usage>INPUT</usage>
        <sequence>3</sequence>
        <isGroupByField>false</isGroupByField>
    </decisionTableParameters>
    <decisionTableParameters>
        <fieldName>Name</fieldName>
        <operator>Equals</operator>
        <usage>INPUT</usage>
        <sequence>4</sequence>
        <isGroupByField>false</isGroupByField>
    </decisionTableParameters>
    <decisionTableSourceCriterias>
        <sourceFieldName>IsDeleted</sourceFieldName>
        <operator>Equals</operator>
        <value>false</value>
        <sequenceNumber>1</sequenceNumber>
        <valueType>Literal</valueType>
    </decisionTableSourceCriterias>
    <description>Sample DT created for md-common tests</description>
    <filterResultBy>UniqueValues</filterResultBy>
    <setupName>Sample DT</setupName>
    <sourceObject>WorkBadgeDefinition</sourceObject>
    <sourceConditionLogic>1</sourceConditionLogic>
    <status>Draft</status>
    <type>LowVolume</type>
    <usageType>Bre</usageType>
    <doesConsiderNullValue>false</doesConsiderNullValue>
    <refreshStatus>Failed</refreshStatus>
    <refreshFailureReason>Failed due to limit violation.</refreshFailureReason>
    <executionType>Hbpo</executionType>
    <lastIncrementalSyncDate>""</lastIncrementalSyncDate>
```

```
    <uploadStatus>Completed</uploadStatus>
    <isIncrementalSyncEnabled>false</isIncrementalSyncEnabled>
    <hasIncrementalSyncFailed>false</hasIncrementalSyncFailed>
</DecisionTable>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
   <fullName>Sample DT Package</fullName>
   <description>Package created for md-common tests</description>
   <types>
      <members>Sample_DT</members>
      <name>DecisionTable</name>
   </types>
   <types>
      <members>DSL_Sample</members>
      <members>Sample_DT_Default</members>
      <name>DecisionTableDatasetLink</name>
   </types>
   <version>51.0</version>
</Package>
```

DecisionTableDatasetLink

Represents the information about a dataset link associated with a decision table. In a dataset link, select an object for whose records, the decision table must provide an outcome.This type extends the Metadata metadata type and inherits its `fullName` field.

Note: Dataset links are supported only for Standard decision tables.

File Suffix and Directory Location

DecisionTableDatasetLink components have the suffix `.decisionTableDatasetLink` and are stored in the `decisionTableDatasetLinks` folder.

Version

DecisionTableDatasetLink components are available in API version 51.0 and later.

Special Access Rules

To use this metadata type, your Salesforce org must have the Loyalty Management or the Rebate Management license.

Fields

Field Name	Field Type	Description
`decisionTableName`	string	Required. The name of the associated decision table.

Field Name	Field Type	Description
decisionTblDatasetParameters	DecisionTblDatasetParameters	Mapping between a decision table parameter and a field of the object selected in the dataset link.
description	string	The description of the dataset link.
isDefault	boolean	Indicates whether a dataset link is the default dataset link for a decision table.
setupName	string	Required. The name of the decision table dataset link, which appears in Setup.
sourceObject	string	Required. The name of the object being evaluated.

DecisionTblDatasetParameters

Represents the mapping between a decision table parameter and a field of the object selected in the dataset link.

The mapping allows the decision table to know which object fields must be compared to the input-output fields of the decision table.

Fields

Field Name	Field Type	Description
datasetFieldName	string	Required. Name of the dataset field whose value must be compared against an Input type decision table parameter when providing the outcome.
fieldName	string	Required. The API name of the decision table field that is selected as an input or output for the decision table dataset link.

Declarative Metadata Sample Definition

The following is an example of a DecisionTableDatasetLink component.

```
<?xml version="1.0" encoding="UTF-8"?>
<DecisionTableDatasetLink xmlns="http://soap.sforce.com/2006/04/metadata">
   <decisionTableName>Sample_DT</decisionTableName>
   <decisionTblDatasetParameters>
      <fieldName>IsDeleted</fieldName>
      <datasetFieldName>IsDeleted</datasetFieldName>
   </decisionTblDatasetParameters>
   <decisionTblDatasetParameters>
      <fieldName>LimitNumber</fieldName>
      <datasetFieldName>CallDurationInSeconds</datasetFieldName>
   </decisionTblDatasetParameters>
   <decisionTblDatasetParameters>
      <fieldName>Name</fieldName>
      <datasetFieldName>Subject</datasetFieldName>
   </decisionTblDatasetParameters>
   <description>DSL created for md-common tests</description>
```

```
    <isDefault>false</isDefault>
    <sourceObject>Task</sourceObject>
    <setupName>DSL Sample</setupName>
</DecisionTableDatasetLink>
```

The following is an example of a default DecisionTableDatasetLink component.

```
<?xml version="1.0" encoding="UTF-8"?>
<DecisionTableDatasetLink xmlns="http://soap.sforce.com/2006/04/metadata">
    <decisionTableName>Sample_DT</decisionTableName>
    <isDefault>true</isDefault>
    <sourceObject>WorkBadgeDefinition</sourceObject>
    <setupName>Default DSL Sample</setupName>
</DecisionTableDatasetLink>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <fullName>Sample DT Package</fullName>
    <description>Package created for md-common tests</description>
    <types>
        <members>Sample_DT</members>
        <name>DecisionTable</name>
    </types>
    <types>
        <members>DSL_Sample</members>
        <members>Sample_DT_Default</members>
        <name>DecisionTableDatasetLink</name>
    </types>
    <version>51.0</version>
</Package>
```

DecisionMatrixDefinition

Represents a definition of a decision matrix.

Note: Before deploying a decision matrix or a decision matrix version to a target org, review these decision matrix migration considerations.

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

DecisionMatrixDefinition components have the suffix `.decisionMatrixDefinition` and are stored in the `decisionMatrixDefinition` folder.

Version

DecisionMatrixDefinition components are available in API version 55.0 and later.

Fields

Field Name	Description
`description`	**Field Type** string **Description** Describes a decision matrix definition.
`groupKey`	**Field Type** string **Description** A key for grouping matrix rows in different versions, such as a geographic region or a product code.
`label`	**Field Type** string **Description** Required. The UI label of a decision matrix definition.
`processType`	**Field Type** ExpsSetProcessType (enumeration of type string) **Description** The process type that uses the expression set rule. Valid values are: • `Bre` • `GpaCalculation` • `PlanCostCalculation` • `RatingDiscovery` • `StudentSuccess` Note: When Business Rules Engine is enabled for a Salesforce instance, the default value is '`Bre`'. Other usage types may be available to you depending on your industry solution and permission sets. Available in API version 59.0 and later.
`subGroupKey`	**Field Type** string **Description** A subgroup key for grouping matrix rows in different versions, such as a geographic region or a product code. For example, if the `groupKey` is `Country`, the `subGroupKey` can be `State` or `Province`.

Field Name	Description
`type`	**Field Type** DecisionMatrixType (enumeration of type string) **Description** The type of a decision matrix. Valid values are: • `Grouped` • `Standard`
`versions`	**Field Type** DecisionMatrixDefinitionVersion[] **Description** Represents an array of decision matrix version definitions in a decision matrix. This array must contain at least one version.

DecisionMatrixDefinitionVersion

Represents a definition of a decision matrix version.

Field Name	Description
`columns`	**Field Type** DecisionMatrixDefinitionVersionColumn[] **Description** Represents an array of columns in a decision matrix definition version.
`decisionMatrixDefinition`	**Field Type** string **Description** The full name of a decision matrix version.
`endDate`	**Field Type** dateTime **Description** The date until which a decision matrix definition version is available for use.
`groupKeyValue`	**Field Type** string **Description** The value of the `groupKey` for a decision matrix definition version. For example, if the `groupKey` is `Country`, the `groupKeyValue` can be `United States`.

Field Name	Description
`label`	**Field Type** string **Description** Required. The UI label of a decision matrix definition version.
`rank`	**Field Type** int **Description** The rank of the `Decision Matrix Definition Version`. When more than one enabled version matches a decision matrix call, and the start date time to end date time spans overlap, the version with the highest rank is chosen. Available in API version 64.0 and later.
`startDate`	**Field Type** dateTime **Description** Required. The date from when a decision matrix definition version is available for use.
`status`	**Field Type** DecisionMatrixDefStatus (enumeration of type string) **Description** Required. Specifies the status of a decision matrix definition version. Valid values are: • `Active` • `Draft` • `Inactive` • `InvalidDraft` • `Obsolete`
`subGroupKeyValue`	**Field Type** string **Description** The value of the subgroup key for a decision matrix definition version. For example, if the `subGroupKey` is `State` or `Province`, the `subGroupKeyValue` can be `California`.

Field Name	Description
versionNumber	**Field Type** int **Description** Required. The version number of a decision matrix definition.

DecisionMatrixDefinitionVersionColumn

Represents a definition of a column in a decision matrix definition version.

Field Name	Description
columnType	**Field Type** DecisionMatrixColumnType (enumeration of type string) **Description** Required. Specifies whether a column is for an input or output. Valid values are: • Input • Output
dataType	**Field Type** DecisionMatrixDataType (enumeration of type string) **Description** Required. The type of data that's stored in a column. Valid values are: • Boolean • Currency • Number • NumberRange • Percent • Text • TextRange
displaySequence	**Field Type** int **Description** Required.

Field Name	Description
	Represents the position of a column in the column order.
`isWildcardColumn`	**Field Type** boolean **Description** Required. Specifies whether a column stores a wildcard value (`true`) or not (`false`). The default value is `false`.
`name`	**Field Type** string **Description** Required. The full name of a decision matrix definition version column.
`rangeValue`	**Field Type** string **Description** A list of values that define range boundaries.
`wildcardValue`	**Field Type** string **Description** The wildcard value such as `ALL`.

Declarative Metadata Sample Definition

The following is an example of a DecisionMatrixDefinition component.

```
<?xml version="1.0" encoding="UTF-8"?>
<DecisionMatrixDefinition
 xmlns="http://soap.sforce.com/2006/04/metadata">
 <label>HealthCloudUM_ValidRegions</label>
 <type>Standard</type>
 <versions>
  <fullName>HealthCloudUM_ValidRegions_V1</fullName>
  <columns>
   <columnType>Input</columnType>
   <dataType>Text</dataType>
   <displaySequence>2</displaySequence>
   <isWildcardColumn>false</isWildcardColumn>
   <name>State</name>
  </columns>
  <columns>
```

```
    <columnType>Input</columnType>
    <dataType>Text</dataType>
    <displaySequence>1</displaySequence>
    <isWildcardColumn>false</isWildcardColumn>
    <name>City</name>
   </columns>
   <columns>
    <columnType>Output</columnType>
    <dataType>Boolean</dataType>
    <displaySequence>3</displaySequence>
    <isWildcardColumn>false</isWildcardColumn>
    <name>IsValid</name>
   </columns>
   <decisionMatrixDefinition>HealthCloudUM_ValidRegions</decisionMatrixDefinition>
   <label>HealthCloudUM_ValidRegions V1</label>
   <startDate>2022-05-02T13:04:06.000Z</startDate>
   <status>Draft</status>
   <versionNumber>1</versionNumber>
  </versions>
</DecisionMatrixDefinition>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package
 xmlns="http://soap.sforce.com/2006/04/metadata">
 <types>
  <members>*</members>
  <name>DecisionMatrixDefinition</name>
 </types>
 <version>64.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

DelegateGroup

Represents a group of users who have the same administrative privileges. These groups are different from public groups used for sharing.

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

DelegateGroup components have the suffix `.delegateGroup` and are stored in the `delegateGroups` folder. The file prefix must match the developer name of the delegate group. For example, a delegate group with a developer name of MyDelegateGroup would have a file name of `MyDelegateGroup.delegateGroup`.

Version

DelegateGroup components are available in API version 36.0 and later.

Special Access Rules

Only users with the "View Setup and Configuration" permission can be delegated administrators. As of Spring '20 and later, only users with "View Setup" or "Configuration" permission can access this object.

Fields

Field Name	Field Type	Description
`customObjects`	string[]	The custom objects associated with the group. Delegated administrators can customize nearly every aspect of each of those custom objects, including creating a custom tab. However, they can't create or modify relationships on the objects or set organization-wide sharing defaults. Delegated administrators must have access to custom objects to access the merge fields on those objects from formulas.
`groups`	string[]	The groups with users assigned by delegated administrators.
`label`	string	Required. The delegated group's non-API name.
`loginAccess`	boolean	Required. Allows users in this group to log in as users in the role hierarchy that they administer (`true`) or not (`false`). Depending on your organization settings, individual users must grant login access to allow their administrators to log in as them.
`permissionSetGroups`	string[]	The permission set groups that can be assigned to users in specified roles and all subordinate roles by delegated administrators.
`permissionSets`	string[]	The permission sets that can be assigned to users in specified roles and all subordinate roles by delegated administrators.
`profiles`	string[]	The profiles that can be assigned to users by delegated administrators.
`roles`	string[]	The roles and subordinates for which delegated administrators of the group can create and edit users.

Declarative Metadata Sample Definition

The following is an example of a DelegateGroup component.

```
<?xml version="1.0" encoding="UTF-8"?>
<DelegateGroup xmlns="http://soap.sforce.com/2006/04/metadata">
    <label>MyDelegateGroup</label>
    <loginAccess>true</loginAccess>
    <name>MyDelegateGroup</name>
    <profiles>Chatter Free User</profiles>
    <profiles>Chatter Moderator User</profiles>
```

```
    <profiles>Marketing User</profiles>
    <permissionSetGroups>My Permission Set Group</permissionSetGroups>
    <permissionSets>My Permset</permissionSets>
    <roles>LesserBossMan</roles>
</DelegateGroup>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>DelegateGroup</name>
    </types>
    <version>64.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

DigitalExperienceBundle

Represents a text-based code structure of your organization's workspaces, organized by workspace type, and each workspace's content items.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

DigitalExperienceBundle components have the suffix `.digitalExperience` and are stored in the `digitalExperiences` folder.

DigitalExperienceBundle uses workspaces and content types to organize your data in a content-focused structure.

- Workspace: For enhanced Lightning Web Runtime (LWR) sites, a collection of related content items that form the site when combined with data from the DigitalExperienceConfig metadata type.

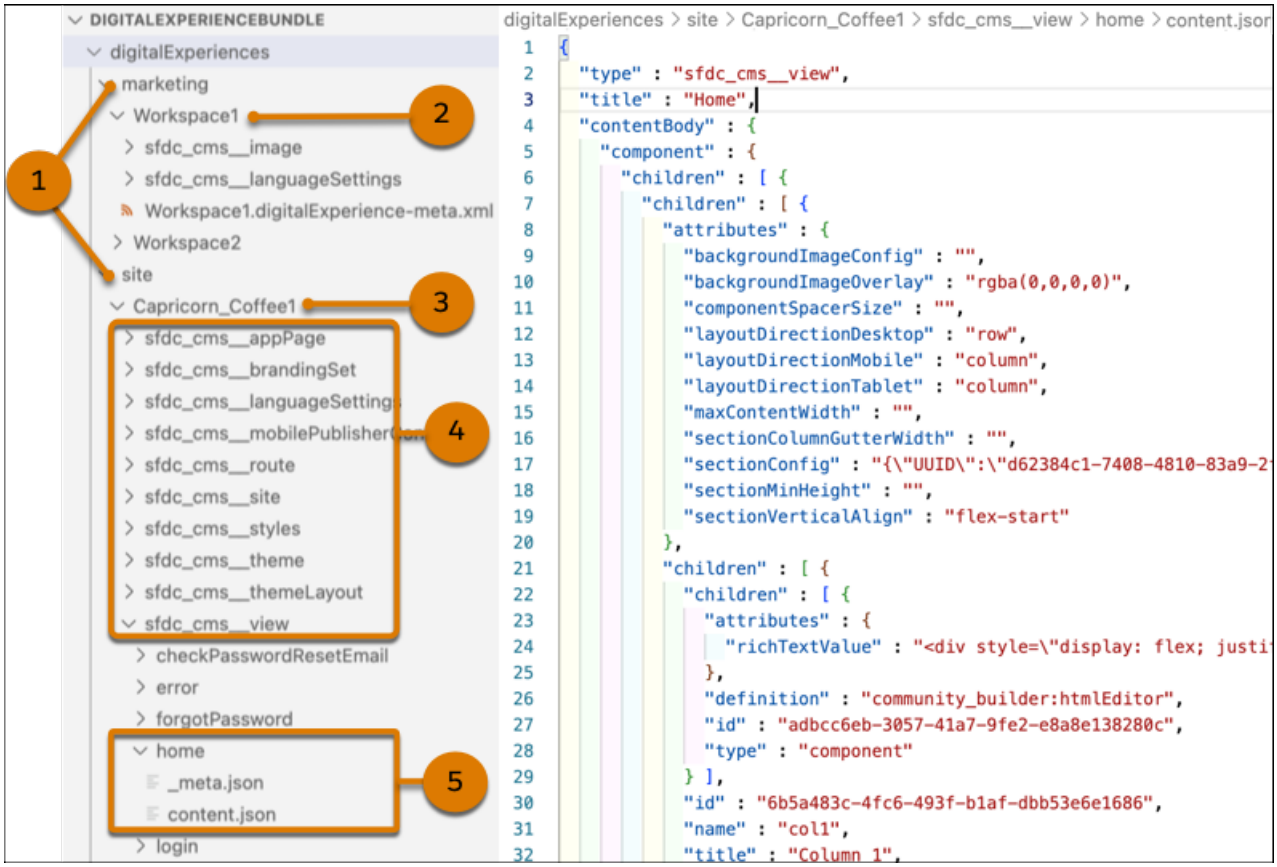
  For Marketing Cloud, a collection of related content items contained in a marketing workspace.

- Workspace type: A way to categorize different kinds of workspaces. For example, the workspace type for enhanced LWR sites is `site`, and the workspace type for marketing workspaces in Marketing Cloud is `marketing`. The workspace type determines which content types are available in the workspace. In the DigitalExperienceBundle folder structure, all workspaces of a given type are under that workspace type. `site` and `marketing` are the supported workspace types.
- Content types: A way to categorize different kinds of content in a workspace. For example, all routes in an enhanced LWR site are stored under a content type folder called `sfdc_cms__route`. Similarly, forms for a marketing workspace are stored under a content type folder called `sfdc_cms__form`.

- Content items: For enhanced LWR sites, the individual settings and site components that make up an enhanced LWR site. For example, each of the routes in an enhanced LWR site is a single content item.

  For marketing workspaces, the content items used in marketing campaigns. For example, each form in a workspace is a single content item.

Here's an example of the DigitalExperienceBundle structure.



When retrieved, DigitalExperienceBundle contains workspace type folders (1) under the digitalExperiences folder.

The marketing folder contains one or more workspace folders (2), each representing a marketing workspace in Marketing Cloud. The site folder contains one or more workspace folders (3), each representing the workspace for an individual enhanced LWR site. Each workspace folder contains an `XML` file with information about the workspace, such as the label. For enhanced LWR sites, be sure to keep the label value in sync with the site's network name.

Each workspace folder also contains several content type folders that represent each of the different content types (4) used in that workspace. For example, marketing workspaces support landing pages, forms, emails, and referenced images and branding.

Finally, each content type folder can contain one or more content subfolders. Each content subfolder can contain additional subfolders and several files that, when combined, represent an individual content item, such as a specific view (5).

- A `_meta.json` file that contains the metadata for the content item. Use the `_meta.json` file to learn the location of a content item within the workspace, or to move the content item to another location, including creating a new location for the content item. You can also use the `_meta.json` file to view a content item's parent-child relationships, to move the content item from one parent to another, or to remove a parent-child relationship entirely.

- A `content.json` file that contains the primary version of the content item. Each `content.json` file includes values for the content item's type, title, and content body. Use this file to edit the content's properties on your local machine or scratch org and then deploy.
- If applicable, additional `JSON` files that represent variants of the content item, such as language translations.

The `_meta.json` file contains several properties:

Property	Description
`apiName`	**Field Type** string **Description** Required. The name of the associated content item. The `apiName` allows only alphanumeric and underscore characters and must begin with a letter. You can use an `apiName` one time per content type in a given workspace. For example, a single workspace can contain both a view named "home" and a route named "home" but can't contain two views named "home".
`parent`	**Field Type** string **Description** The developer name of the content item's parent. If the content item doesn't have a parent, then either the parent value is blank or the parent property isn't displayed at all.
`path`	**Field Type** string **Description** The location of the content item within the workspace's folder structure. The value is blank for content types at the root level, such as `appPage` and `site`.
`type`	**Field Type** string **Description** Required. The full name of the content type from which the content item was created. The type is prefixed with `sfdc_cms__`. For example, all views in a site workspace have the content type `sfdc_cms__view`.

Version

DigitalExperienceBundle components are available in API version 56.0 and later.

Special Access Rules

In Experience Cloud, can use DigitalExperienceBundle for enhanced LWR sites created in Winter '23 or later. For Aura sites and other LWR sites, use the ExperienceBundle (recommended) or the SiteDotCom on page 2179 metadata types.

In Marketing Cloud, you must have a contributor role in a marketing workspace to retrieve it. Only marketing workspaces are supported when using DigitalExperienceBundle. You can't package or deploy general workspaces.

Fields

Field Name	Description
`description`	**Field Type** string **Description** Contains the description of the workspace. For site workspaces, this value is empty.
`digitalExperienceFolderShares`	**Field Type** DigitalExperienceFolderShare[] **Description** The list of folders in the source marketing workspace that are shared with target marketing workspaces. Available in API version 61.0 and later.
`label`	**Field Type** string **Description** Required. A user-friendly name for DigitalExperienceBundle, which is defined when the DigitalExperienceBundle is created.
`spaceResources`	**Field Type** DigitalExperience[] **Description** The list of resources in this DigitalExperienceBundle. Each resource represents a content type, such as views, routes, themes, and languageSettings.

DigitalExperience

Represents content in the bundle. When retrieved as part of DigitalExperienceBundle, DigitalExperience represents all content for the requested workspace or workspaces. When retrieved on its own, DigitalExperience represents only the content types you specify.

This subtype extends the MetadataWithContent metadata type and inherits its `content` and `fullName` fields.

When you retrieve DigitalExperience, the folder structure matches that of DigitalExperienceBundle, with only the specified content returned.

Field Name	Description
`fileName`	**Field Type** string **Description** Required. Name of the resource file.
`filePath`	**Field Type** string **Description** Path to the file within the artifact folder.
`format`	**Field Type** string **Description** Required. Only `JSON` is allowed.

DigitalExperienceFolderShare

Represents a folder in a source marketing workspace that's shared with other target marketing workspaces. Available in API version 61.0 and later.

Field Name	Description
`folderPath`	**Field Type** string **Description** The root folder of the shared workspace. The allowed value is `_root`.
`sharedWith`	**Field Type** SharedWith[] **Description** The list of target workspaces that the source workspace is shared with.

SharedWith

Represents a target marketing workspace that the source marketing workspace is shared with. Available in API version 61.0 and later.

Field Name	Description
`fullyQualifiedName`	**Field Type** string **Description** The target workspace that the source workspace is shared with. It uses the format *`workspace_type/target_workspace_name`*. For example, `marketing/Workspace2`.

Declarative Metadata Sample Definition

The following is an example of a DigitalExperienceBundle component.

```
<?xml version="1.0" encoding="UTF-8"?>
<DigitalExperienceBundle xmlns="http://soap.sforce.com/2006/04/metadata">
    <description>content</description>
<label>isv1</label>
</DigitalExperienceBundle>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>site/isv1</members>
        <name>DigitalExperienceBundle</name>
    </types>
    <version>56.0</version>
</Package>
```

Usage

Tip: Before you update the `JSON` files of an Experience Builder site, we recommend making a copy of the site's folder as a backup.

To retrieve and deploy DigitalExperienceBundle, use legacy sfdx commands.

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

DigitalExperienceBundle: Marketing Workspace Bundle and Folders

DigitalExperienceBundle uses the `marketing` workspace type to organize content items used in marketing campaigns in a content-focused, text-based code structure.

DigitalExperienceBundle: Site Workspace Bundle and Folders

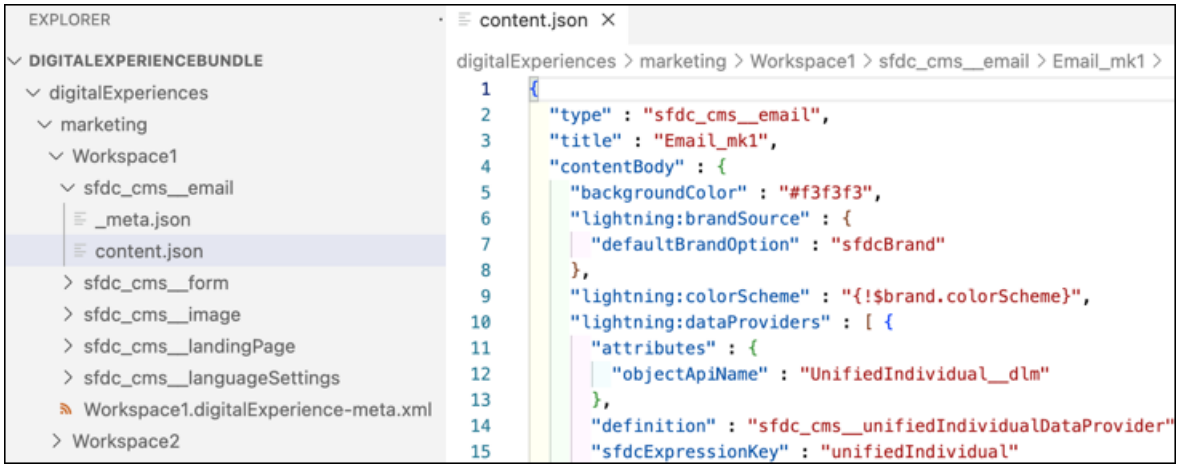
DigitalExperienceBundle uses the `site` workspace type to organize data for enhanced LWR sites in a content-focused, text-based code structure.

DigitalExperienceBundle: Marketing Workspace Bundle and Folders

DigitalExperienceBundle uses the `marketing` workspace type to organize content items used in marketing campaigns in a content-focused, text-based code structure.

For Marketing Cloud, the `marketing` folder contains one or more workspace folders, each representing an individual marketing workspace. Each workspace folder contains a collection of related content items, such as landing pages, forms, and emails, and their associated images and branding.

The folder for each marketing workspace includes content type folders, content item subfolders, and associated data that's contained in `content.json` and `_meta.json` files.



The following content type folders represent the content types that are supported in a marketing workspace. For example, forms for a marketing workspace are stored under a content type folder called `sfdc_cms__form`.

- sfdc_cms__brand Folder
- sfdc_cms__brandSettings Folder
- sfdc_cms__email Folder
- sfdc_cms__form Folder
- sfdc_cms__image Folder
- sfdc_cms__landingPage Folder
- sfdc_cms__languageSettings Folder

sfdc_cms__brand Folder

This content type folder contains one content subfolder per brand. Each content subfolder contains two or more `JSON` files:

- `_meta.json`
- `content.json`
- If applicable, additional `JSON` files that represent variations of the content item

<apiName>/content.json

```
{
  "type" : "sfdc_cms__brand",
  "title" : "brand 1",
  "contentBody" : {
    "baseFontFamily" : "{!$brand.fontFamily.arial}",
    "baseFontSize" : {
      "unit" : "px",
      "value" : 16.0
    },
    "borderRadius" : {
      "round" : {
        "unit" : "rem",
        "value" : 0.25
      },
      "square" : {
        "unit" : "rem",
        "value" : 0.0
      }
    },
    "borderWeight" : {
      "medium" : {
        "unit" : "rem",
        "value" : 0.125
      },
      "none" : {
        "unit" : "rem",
        "value" : 0.0
      },
      ...
    },
    "buttonStyleGroup" : {
      "primary" : {
        "lightning:borderRadius" : "{!$brand.borderRadius.round}",
        "lightning:borderWidth" : "{!$brand.borderWeight.thin}",
        "lightning:buttonColorGroup" : {
          "backgroundColor" : "{!$brand.colorScheme.primaryAccent}",
          "backgroundHoverColor" : "{!$brand.colorScheme.primaryAccentDerived}",
          "borderColor" : "{!$brand.colorScheme.primaryAccent}",
          "borderHoverColor" : "{!$brand.colorScheme.primaryAccentDerived}",
          "textColor" : "{!$brand.colorScheme.primaryAccentContrast}",
          "textHoverColor" : "{!$brand.colorScheme.primaryAccentContrastDerived}"
        },
        "lightning:padding" : {
          "bottom" : {
            "unit" : "rem",
            "value" : 0.5
          },
          ...
        },
        "lightning:typography" : "{!$brand.typography.button.button1}"
      },
      "secondary" : {...},
      "tertiary" : {...}
```

```
  },
  "colorScheme" : {
    "contrast" : "#000000",
    "neutral" : "#747474",
    "primaryAccent" : "#99F077",
    "primaryAccentContrast" : "#ffffff",
    "primaryAccentContrastDerived" : "#000000",
    "primaryAccentDerived" : "#7fd65f",
    "root" : "#ffffff"
  },
  "fontFamily" : {
    "arial" : {
      "category" : "sans-serif",
      "fallbacks" : [ "Helvetica" ],
      "name" : "Arial"
    },
    "arialBlack" : {
      "category" : "sans-serif",
      "fallbacks" : [ "Gadget" ],
      "name" : "Arial Black"
    },
    ...
  },
  "fontSize" : {
    "large" : {
      "unit" : "rem",
      "value" : 1.125
    },
    "medium" : {
      "unit" : "rem",
      "value" : 1.0
    },
    ...
  },
  "spacing" : {
    "large" : {
      "bottom" : {
        "unit" : "rem",
        "value" : 1.5
      },
      "left" : {
        "unit" : "rem",
        "value" : 1.5
      },
      "right" : {
        "unit" : "rem",
        "value" : 1.5
      },
      "top" : {
        "unit" : "rem",
        "value" : 1.5
      }
    },
    ...
```

```
    },
    "typography" : {
      "button" : {
        "button1" : {
          "fontFamily" : "{!$brand.baseFontFamily}",
          "fontSize" : "{!$brand.fontSize.medium}",
          "fontWeight" : "{!$brand.fontWeight.normal}",
          "letterSpacing" : "normal",
          "lineHeight" : 1.5,
          "textTransform" : "none"
        }
      },
      ...
    },
    ...
    "lightning:dataProviders" : [ ],
    "sfdc_cms:einsteinBrandProperties" : {
      "personality" : {
        "defaultPersonality" : "professional"
      }
    },
    "sfdc_cms:variants" : [ ]
  },
  "urlName" : "brand-1",
  "sfdc_cms:title" : "brand 1",
}
```

sfdc_cms__brandSettings Folder

This content type folder contains one content subfolder called brandSettings. The brandSettings content subfolder contains two or more `JSON` files:

- `_meta.json`
- `content.json`
- If applicable, additional `JSON` files that represent variations of the content item

<apiName>`/content.json`

```
{
  "type" : "sfdc_cms__brandSettings",
  "title" : "Brand Settings",
  "contentBody" : {
    "defaultBrand" : "brand3"
  },
  "urlName" : "brand-settings"
}
```

sfdc_cms__email Folder

This content type folder contains one content subfolder per email. Each content subfolder contains two or more `JSON` files:

- `_meta.json`
- `content.json`

- If applicable, additional `JSON` files that represent variations of the content item

Note: In marketing workspaces, the default data graph, personalization recommenders, personalization points, and decisions aren't included in the bundle. If the workspace includes emails with personalized content that's based on these objects, then:

- Any merge field or repeater that uses the default data graph or a personalization recommender from the source org is broken in the target org.
- Any dynamic content variations of email components are removed and only the default variations appear in the email.

<apiName>/content.json

```
{
  "type" : "sfdc_cms__email",
  "title" : "Email_marketingSpaceA",
  "contentBody" : {
    "backgroundColor" : "#f3f3f3",
    "lightning:brandSource" : {
      "defaultBrandOption" : "sfdcBrand"
    },
    "lightning:colorScheme" : "{!$brand.colorScheme}",
    "lightning:dataProviders" : [ {
      "attributes" : {
        "objectApiName" : "UnifiedIndividual__dlm"
      },
      "definition" : "sfdc_cms__unifiedIndividualDataProvider",
      "sfdcExpressionKey" : "unifiedIndividual"
    } ],
    "lightning:padding" : "{!$brand.spacing.none}",
    "messagePurpose" : "promotional",
    "sfdc_cms:block" : {
      "definition" : "sfdc_cms/rootContentBlock",
      "id" : "6458e24b-c1a8-4f7d-b6f0-3659c092f1c3",
      "type" : "block",
      "children" : [ {
        "attributes" : {
          "lightning:borderRadius" : "{!$brand.borderRadius.square}",
          "lightning:borderWidth" : "{!$brand.borderWeight.none}",
          "lightning:colorScheme" : "{!$brand.colorScheme}",
          "lightning:margin" : "{!$brand.spacing.none}",
          "lightning:padding" : "{!$brand.spacing.xSmall}",
          "stackOnMobile" : true,
          "lightning:backgroundImage" : {
            "repeat" : "no-repeat",
            "position" : "center center",
            "size" : "cover"
          }
        },
        "definition" : "lightning/section",
        "id" : "b61c4d08-7985-41f2-a38c-7f8338e56e00",
        "type" : "block",
        "children" : [ {
          "attributes" : {
            "columnWidth" : 12.0,
            "lightning:borderRadius" : "{!$brand.borderRadius.square}",
            "lightning:borderWidth" : "{!$brand.borderWeight.none}",
```

```
            "lightning:colorScheme" : "{!$brand.colorScheme}",
            "lightning:margin" : "{!$brand.spacing.none}",
            "lightning:padding" : "{!$brand.spacing.xSmall}",
            "lightning:verticalAlignment" : "top",
            "lightning:backgroundImage" : {
              "repeat" : "no-repeat",
              "position" : "center center",
              "size" : "cover"
            }
          },
          "definition" : "lightning/column",
          "id" : "778d9976-82ec-49aa-a3de-ac6485332434",
          "type" : "block",
          "children" : [ ]
        } ]
      } ]
    },
    "sfdc_cms:title" : "Email_marketingSpaceA",
    "subjectLine" : "Email_marketingSpaceA subject{!$organization.Address}",
    "lightning:expressions" : [ ],
    "lightning:backgroundImage" : {
      "repeat" : "no-repeat",
      "position" : "center center",
      "size" : "cover"
    },
    "sfdc_cms:variants" : [ ]
  },
  "urlName" : "email-mk1"
}
```

sfdc_cms__form Folder

This content type folder contains one content subfolder per form. Each content subfolder contains two or more `JSON` files:

- `_meta.json`
- `content.json`
- If applicable, additional `JSON` files that represent variations of the content item

<apiName>`/content.json`

```
{
  "type" : "sfdc_cms__form",
  "title" : "Form1_mk1",
  "contentBody" : {
    "lightning:brandSource" : {
      "defaultBrandOption" : "sfdcBrand"
    },
    "lightning:dataProviders" : [ {
      "attributes" : {
        "objectApiName" : "Account",
        "recordTypeId" : "012000000000000AAA"
      },
      "definition" : "sfdc_cms__recordDataProvider",
      "sfdcExpressionKey" : "Flow1"
```

```
  } ],
  "sfdc_cms:block" : {
    "definition" : "sfdc_cms/rootContentBlock",
    "id" : "fef7b2b0-5ddf-4f0e-b0d5-cdbc77a897e9",
    "type" : "block",
    "children" : [ {
      "attributes" : {
        "lightning:borderRadius" : "{!$brand.borderRadius.square}",
        "lightning:borderWidth" : "{!$brand.borderWeight.none}",
        "lightning:colorScheme" : "{!$brand.colorScheme}",
        "lightning:margin" : "{!$brand.spacing.none}",
        "lightning:padding" : "{!$brand.spacing.xSmall}",
        "stackOnMobile" : true,
        "lightning:backgroundImage" : {
          "repeat" : "no-repeat",
          "position" : "center center",
          "size" : "cover"
        }
      },
      "definition" : "lightning/section",
      "id" : "43dc4273-47e2-43ad-9e64-f0862eb0fcdf",
      "type" : "block",
      "children" : [ {
        "attributes" : {
          "columnWidth" : 12.0,
          "lightning:borderRadius" : "{!$brand.borderRadius.square}",
          "lightning:borderWidth" : "{!$brand.borderWeight.none}",
          "lightning:colorScheme" : "{!$brand.colorScheme}",
          "lightning:margin" : "{!$brand.spacing.none}",
          "lightning:padding" : "{!$brand.spacing.xSmall}",
          "lightning:verticalAlignment" : "top",
          "lightning:backgroundImage" : {
            "repeat" : "no-repeat",
            "position" : "center center",
            "size" : "cover"
          }
        },
        "definition" : "lightning/column",
        "id" : "95fc1b5c-481d-4d32-bd03-fec0a4d7aaa0",
        "type" : "block",
        "children" : [ {
          "attributes" : {
            "lightning:borderRadius" : "{!$brand.borderRadius.square}",
            "lightning:borderWidth" : "{!$brand.borderWeight.none}",
            "lightning:formInputColorGroup" : {
              "backgroundColor" : "{!$brand.colorScheme.root}",
              "borderColor" : "{!$brand.colorScheme.neutral}",
              "textColor" : "{!$brand.colorScheme.contrast}"
            },
            "lightning:horizontalAlignment" : "left",
            "lightning:inputTypography" : "{!$brand.typography.input.input1}",
            "lightning:labelTypography" : "{!$brand.typography.label.label1}",
            "lightning:margin" : "{!$brand.spacing.none}",
            "lightning:padding" : "{!$brand.spacing.none}",
```

```
              "maxLength" : 255.0,
              "sfdc_cms:fieldReference" : "{!Flow1.Name}",
              "sfdc_cms:formInputLabelProperty" : "Account Name",
              "sfdc_cms:formInputNameProperty" : "Name",
              "sfdc_cms:formInputRequiredProperty" : true,
              "width" : "auto"
            },
            "definition" : "lightning/inputText",
            "id" : "6aac0596-26c6-457a-9a9a-cc43ba622739",
            "type" : "block"
          } ]
        } ]
      }, {
        "attributes" : {
          "lightning:borderRadius" : "{!$brand.borderRadius.square}",
          "lightning:borderWidth" : "{!$brand.borderWeight.none}",
          "lightning:colorScheme" : "{!$brand.colorScheme}",
          "lightning:margin" : "{!$brand.spacing.none}",
          "lightning:padding" : "{!$brand.spacing.xSmall}",
          "stackOnMobile" : true,
          "lightning:backgroundImage" : {
            "repeat" : "no-repeat",
            "position" : "center center",
            "size" : "cover"
          }
        },
        "definition" : "lightning/section",
        "id" : "7fe6298e-8c83-4dac-9596-02c629fdc519",
        "type" : "block",
        "children" : [ {
          "attributes" : {
            "columnWidth" : 12.0,
            "lightning:borderRadius" : "{!$brand.borderRadius.square}",
            "lightning:borderWidth" : "{!$brand.borderWeight.none}",
            "lightning:colorScheme" : "{!$brand.colorScheme}",
            "lightning:margin" : "{!$brand.spacing.none}",
            "lightning:padding" : "{!$brand.spacing.xSmall}",
            "lightning:verticalAlignment" : "top",
            "lightning:backgroundImage" : {
              "repeat" : "no-repeat",
              "position" : "center center",
              "size" : "cover"
            }
          },
          "definition" : "lightning/column",
          "id" : "976bff41-3fa9-4d04-aaf8-3590cb87909f",
          "type" : "block",
          "children" : [ {
            "attributes" : {
              "lightning:borderRadius" :
"{!$brand.buttonStyleGroup.primary.lightning:borderRadius}",
              "lightning:borderWidth" :
"{!$brand.buttonStyleGroup.primary.lightning:borderWidth}",
              "lightning:buttonColorGroup" :
```

```
"{!$brand.buttonStyleGroup.primary.lightning:buttonColorGroup}",
              "lightning:horizontalAlignment" : "center",
              "lightning:margin" : "{!$brand.spacing.none}",
            "lightning:padding" : "{!$brand.buttonStyleGroup.primary.lightning:padding}",

              "lightning:typography" :
"{!$brand.buttonStyleGroup.primary.lightning:typography}",
              "sfdc_cms:styleGroup" : "{!$brand.buttonStyleGroup.primary}",
              "text" : "Submit",
              "width" : "auto",
              "lightning:click" : {
                "actions" : [ {
                  "definition" : "sfdc_cms/customEventAction",
                  "attributes" : {
                    "type" : "formsubmit",
                    "options" : {
                      "bubbles" : true
                    }
                  }
                } ]
              }
            },
            "definition" : "lightning/actionButton",
            "id" : "84c67ba2-fffc-46d1-80af-35e66ae85ef3",
            "type" : "block"
          } ]
        } ]
      } ]
    },
    "sfdc_cms:title" : "Form1_mk1",
    "formsubmission" : {
      "actions" : [ {
        "definition" : "sfdc_cms/umaFormSubmissionAction",
        "attributes" : {
          "formId" : "{!$form.id}",
          "pageReferenceId" : "{!$page.id}",
          "formData" : "{!$form.fields}"
        }
      }, {
        "definition" : "sfdc_cms/showThankYouAction",
        "attributes" : {
          "message" : "Thank you for your submission."
        }
      } ]
    }
  },
  "urlName" : "form1-mk1"
}
```

sfdc_cms__image Folder

This content type folder contains one content subfolder per image. Each content subfolder contains two or more `JSON` files and a `_media` subfolder that contains the image file.

- _meta.json
- content.json
- If applicable, additional JSON files that represent variations of the content item

<apiName>/content.json

```
{
  "type" : "sfdc_cms__image",
  "title" : "Img1_mk1",
  "contentBody" : {
    "sfdc_cms:media" : {
      "source" : {
        "mimeType" : "image/png",
        "ref" : "0sNSB000001rKsr2AE",
        "type" : "file",
        "size" : 538158
      }
    }
  },
  "urlName" : "img1-mk1"
}
```

sfdc_cms__landingPage Folder

This content type folder contains one content subfolder per landing page. Each content subfolder contains two or more JSON files:

- _meta.json
- content.json
- If applicable, additional JSON files that represent variations of the content item

<apiName>/content.json

```
{
  "type" : "sfdc_cms__landingPage",
  "title" : "LandingPageA_marketingSpaceA",
  "contentBody" : {
    "lightning:brandSource" : {
      "defaultBrandOption" : "sfdcBrand"
    },
    "sfdc_cms:block" : {
      "definition" : "sfdc_cms/rootContentBlock",
      "id" : "ac065643-646a-4b1e-b5ed-7eeeed90d0d3",
      "type" : "block",
      "children" : [ {
        "attributes" : {
          "lightning:borderRadius" : "{!$brand.borderRadius.square}",
          "lightning:borderWidth" : "{!$brand.borderWeight.none}",
          "lightning:colorScheme" : "{!$brand.colorScheme}",
          "lightning:margin" : "{!$brand.spacing.none}",
          "lightning:padding" : "{!$brand.spacing.xSmall}",
          "stackOnMobile" : true,
          "lightning:backgroundImage" : {
            "repeat" : "no-repeat",
            "position" : "center center",
```

```
      "size" : "cover"
    }
  },
  "definition" : "lightning/section",
  "id" : "f6371eda-aafc-4164-a18f-284e49071b76",
  "type" : "block",
  "children" : [ {
    "attributes" : {
      "columnWidth" : 12.0,
      "lightning:borderRadius" : "{!$brand.borderRadius.square}",
      "lightning:borderWidth" : "{!$brand.borderWeight.none}",
      "lightning:colorScheme" : "{!$brand.colorScheme}",
      "lightning:margin" : "{!$brand.spacing.none}",
      "lightning:padding" : "{!$brand.spacing.xSmall}",
      "lightning:verticalAlignment" : "top",
      "lightning:backgroundImage" : {
        "repeat" : "no-repeat",
        "position" : "center center",
        "size" : "cover"
      }
    },
    "definition" : "lightning/column",
    "id" : "db82b936-f2d8-4d47-b373-71dff7fc1f1d",
    "type" : "block",
    "children" : [ {
      "attributes" : {
        "imageFitConfig" : {
          "width" : {
            "unit" : "%",
            "value" : 100.0
          }
        },
        "imageInfo" : {
          "altText" : "",
          "overrideAltText" : false,
          "source" : {
            "ref" : "Img1_mk1",
            "type" : "imageReference"
          },
          "url" : "/cms/media/MCWJDAQWY2HREBRENINOZIKNNVNM"
        },
        "lightning:borderRadius" : "{!$brand.borderRadius.square}",
        "lightning:borderWidth" : "{!$brand.borderWeight.none}",
        "lightning:colorGroup" : {
          "backgroundColor" : "{!$brand.colorScheme.root}",
          "borderColor" : "{!$brand.colorScheme.neutral}",
          "linkColor" : "{!$brand.colorScheme.primaryAccent}",
          "textColor" : "{!$brand.colorScheme.contrast}"
        },
        "lightning:horizontalAlignment" : "center",
        "lightning:margin" : "{!$brand.spacing.none}",
        "lightning:padding" : "{!$brand.spacing.none}",
        "lightning:typography" : "{!$brand.typography.paragraph.paragraph1}"
      },
```

```
            "definition" : "lightning/image",
            "id" : "6775db07-8343-420c-918a-0d91c193902d",
            "type" : "block"
          } ]
        } ]
      } ]
    },
    "sfdc_cms:seoProperties" : {
      "isIndexed" : false,
      "title" : "LandingPageA_marketingSpaceA"
    },
    "sfdc_cms:title" : "LandingPageA_marketingSpaceA",
    "lightning:dataProviders" : [ ],
    "lightning:backgroundImage" : {
      "repeat" : "no-repeat",
      "position" : "center center",
      "size" : "cover"
    }
  },
  "urlName" : "lp1-mk1"
}
```

sfdc_cms__languageSettings Folder

This content type folder contains one content subfolder called languages. The languages content subfolder contains two or more `JSON` files:

- `_meta.json`
- `content.json`
- If applicable, additional `JSON` files that represent variations of the content item

<apiName>`/content.json`

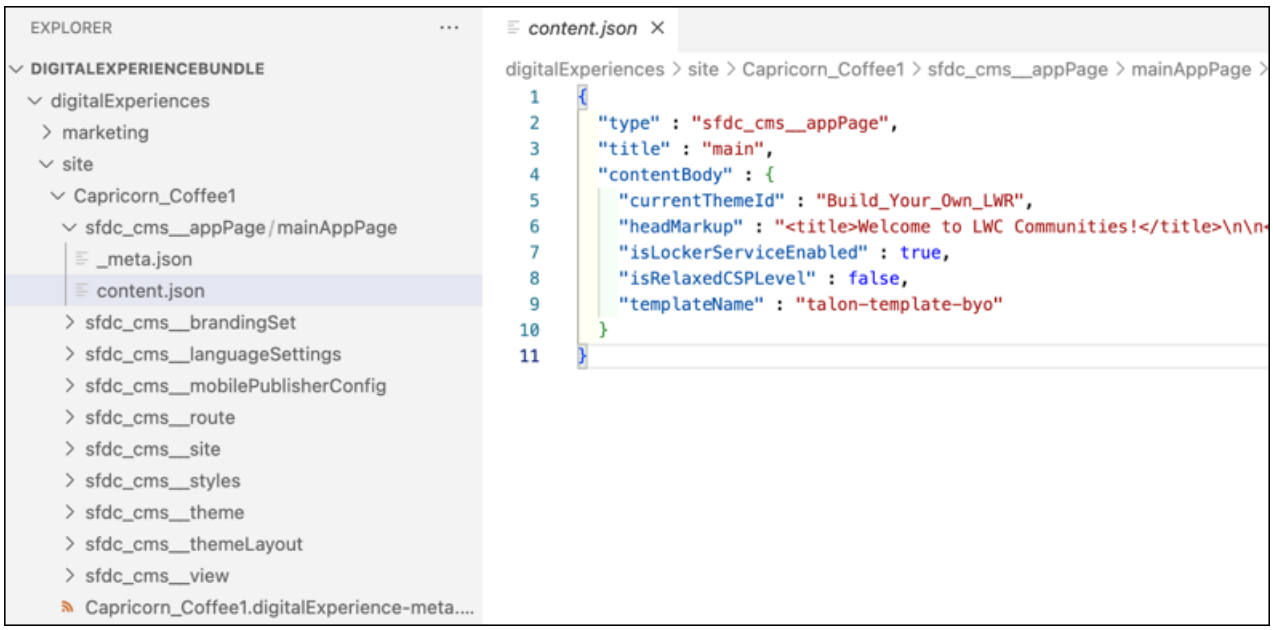
```
{
  "type" : "sfdc_cms__languageSettings",
  "title" : "LanguageContent",
  "contentBody" : {
    "languages" : [ {
      "locale" : "en_US",
      "label" : "English (US)",
      "isActive" : true,
      "isAuthoringOnly" : false
    } ],
    "defaultLocale" : "en_US"
  },
  "urlName" : "languagecontent"
}
```

DigitalExperienceBundle: Site Workspace Bundle and Folders

DigitalExperienceBundle uses the `site` workspace type to organize data for enhanced LWR sites in a content-focused, text-based code structure.

The `site` folder contains one or more workspace folders, each representing the workspace for an individual enhanced LWR site. Each workspace folder contains a collection of related content items, such as settings and site components, that form the site when combined with data from the DigitalExperienceConfig metadata type.

The workspace folder for each site includes content type folders, content item subfolders, and associated data that's contained in `content.json` and `_meta.json` files.



The following content type folders represent the content types that are supported in an enhanced LWR site. For example, all routes in an enhanced LWR site are stored under the `sfdc_cms__route` content type folder.

- sfdc_cms__appPage Folder
- sfdc_cms__brandingSet Folder
- sfdc_cms__languageSettings Folder
- sfdc_cms__route Folder
- sfdc_cms__site Folder
- sfdc_cms__theme Folder
- sfdc_cms__themeLayout Folder
- sfdc_cms__view Folder

sfdc_cms__appPage Folder

This content type folder exists at the root level and contains one content subfolder that represents the site's single-page application. Only one `sfdc_cms__appPage` content item is allowed per site.

The content subfolder contains two or more `JSON` files:

- `_meta.json`
- `content.json`
- If applicable, additional `JSON` files that represent variations of the content item

<apiName>`/content.json`

Property	Description
`currentThemeId`	**Field Type** content link **Description** Represents the `apiName` of the site's active theme.
`headMarkup`	**Field Type** string **Description** Allows the addition of custom markup to the site's main page `<head>` tag. Similar to using **Experience Builder** > **Setting** > **Advanced** > **Head Markup**. See Salesforce Help for markup guidance.
`isLockerServiceEnabled`	**Field Type** boolean **Description** Indicates whether Lightning Locker is enabled (`true`) or not (`false`). The default value is `true`.
`isRelaxedCSPLevel`	**Field Type** boolean **Description** Controls the ability to run scripts and controls script access to third-party hosts. The default is `false`. A `false` value sets a strict CSP security level, and blocks access to inline scripts and all hosts. A `true` value sets a relaxed CSP security level, and permits access to inline scripts and allowed hosts. To learn more, see Select a Security Level in Experience Builder Sites.
`templateName`	**Field Type** string **Description** Required. The unique developer name of the template. Valid values are: • `talon-template-byo`—Represents the Build Your Own (LWR) template • `microsite-template-marketing`—Represents the Microsite template
`title`	**Field Type** string

Property	Description
	Description Required. Represents the label of the appPage.
`type`	**Field Type** string **Description** Required. Represents the content type. The only allowed value is `sfdc_cms__appPage`.

```
{
  "type" : "sfdc_cms__appPage",
  "title" : "main",
  "contentBody" : {
    "currentThemeId" : "Build_Your_Own_LWR",
    "headMarkup" : "<meta charset=\"UTF-8\" />\n<meta name=\"viewport\"
content=\"width=device-width, initial-scale=1\" />\n<title>Welcome to LWC
Communities!</title>\n\n<link rel=\"stylesheet\" href=\"{ basePath
}/assets/styles/styles.css?{ versionKey }\" />\n\n\n<!-- webruntime-branding-shared
stylesheets -->\n<link rel=\"stylesheet\" href=\"{ basePath
}/assets/styles/salesforce-lightning-design-system.min.css?{ versionKey }\" />\n<link
rel=\"stylesheet\" href=\"{ basePath }/assets/styles/dxp-site-spacing-styling-hooks.min.css?{
 versionKey }\" />\n<link rel=\"stylesheet\" href=\"{ basePath
}/assets/styles/dxp-styling-hooks.min.css?{ versionKey }\" />\n<link rel=\"stylesheet\"
href=\"{ basePath }/assets/styles/dxp-slds-extensions.min.css?{ versionKey }\" />\n\n\n<!--
 webruntime-branding-shared stylesheets -->",
    "isLockerServiceEnabled" : true,
    "isRelaxedCSPLevel" : false,
    "templateName" : "talon-template-byo"
  }
}
```

sfdc_cms__brandingSet Folder

This content type folder contains one content subfolder per branding set. Each content subfolder contains two or more `JSON` files:

- `_meta.json`
- `content.json`
- If applicable, additional `JSON` files that represent variations of the content item

<apiName>`/content.json`

Property	Description
`brandingSetType`	**Field Type** enumeration on page 159 of type string **Description** Represents whether the color palette stored in the branding set is for the entire site or a specific section. You can't change one branding set type to another. Valid values are: • `APP`—The branding set applies to the entire site. There can be only one branding set of this type. • `SCOPED`—A `SCOPED` branding set can be applied only to a section component.
`definitionName`	**Field Type** string **Description** Required. Represents the name for the branding set used in the site or template's theme. There are two standard templates that have unique naming: • Build Your Own (LWR) uses talon-template-byo:branding • Microsite uses microsite-template-marketing:branding
`title`	**Field Type** string **Description** Required. Represents the label of the brandingSet content item. The maximum length is 100 characters. The title must be unique within the space's brandingSet content items.
`type`	**Field Type** string **Description** Required. Represents the content type. The only supported value is `sfdc_cms__brandingSet`.
`values`	**Field Type** object (map) **Description** Required.

Property	Description
	Represents a map of branding values that can be applied to a site.


```
{
  "type" : "sfdc_cms__brandingSet",
  "title" : "Build Your Own (LWR)",
  "contentBody" : {
    "brandingSetType" : "APP",
    "definitionName" : "talon-template-byo:branding",
    "values" : {
      "BackgroundColor" : "#ffffff",
      "BaseFontSize" : "1rem",
      "BodyFont" : "Salesforce Sans",
      "BodyFontSize" : "1rem",
      "BodyFontStyle" : "normal",
      "BodyFontWeight" : "400",
      "BodyLetterSpacing" : "0em",
      "BodyLineHeight" : "1.5",
      "BodySmallFont" : "Salesforce Sans",
      "BodySmallFontSize" : "0.75rem",
      "BodySmallFontStyle" : "normal",
      "BodySmallFontWeight" : "400",
      "BodySmallLetterSpacing" : "0em",
      "BodySmallLineHeight" : "1.25",
      "BodySmallTextColor" : "var(--dxp-g-root-contrast)",
      "BodySmallTextDecoration" : "none",
      "BodySmallTextTransform" : "none",
      "BodyTextColor" : "var(--dxp-g-root-contrast)",
      "BodyTextDecoration" : "none",
      "BodyTextTransform" : "none",
      "ButtonActiveColor" : "var(--dxp-s-button-color-1)",
      "ButtonBorderRadius" : "4px",
      "ButtonColor" : "var(--dxp-g-brand)",
      "ButtonFocusColor" : "var(--dxp-s-button-color-1)",
      "ButtonFont" : "Salesforce Sans",
      "ButtonFontSize" : "1rem",
      "ButtonFontStyle" : "normal",
      "ButtonFontWeight" : "400",
      "ButtonHoverColor" : "var(--dxp-s-button-color-1)",
      "ButtonLargeBorderRadius" : "4px",
      "ButtonLargeFontSize" : "1.25rem",
      "ButtonLargePadding" : "1.25rem",
      "ButtonLetterSpacing" : "0em",
      "ButtonLineHeight" : "2",
      "ButtonPadding" : "1rem",
      "ButtonSmallBorderRadius" : "4px",
      "ButtonSmallFontSize" : "0.75rem",
      "ButtonSmallPadding" : "0.75rem",
      "ButtonTextTransform" : "none",
      "ColumnSpacerSizeDesktop" : "1rem",
```

```
"ColumnSpacerSizeMobile" : "0.75rem",
"ComponentSpacerSizeDesktop" : "1.5rem",
"ComponentSpacerSizeMobile" : "1.5rem",
"DropdownBackgroundColor" : "var(--dxp-g-root)",
"DropdownBackgroundHoverColor" : "var(--dxp-g-neutral)",
"DropdownBorderColor" : "var(--dxp-g-neutral)",
"DropdownTextColor" : "var(--dxp-g-root-contrast)",
"DropdownTextHoverColor" : "var(--dxp-g-neutral-contrast)",
"FormElementBackgroundColor" : "var(--dxp-g-root)",
"FormElementBorderColor" : "var(--dxp-g-neutral-3)",
"FormElementBorderRadius" : "4px",
"FormElementBorderWidth" : "1px",
"FormElementLabelColor" : "var(--dxp-g-root-contrast)",
"FormElementTextColor" : "var(--dxp-g-root-contrast)",
"HeadingExtraLargeColor" : "var(--dxp-g-root-contrast)",
"HeadingExtraLargeFont" : "Salesforce Sans",
"HeadingExtraLargeFontSize" : "2.5rem",
"HeadingExtraLargeFontStyle" : "normal",
"HeadingExtraLargeFontWeight" : "300",
"HeadingExtraLargeLetterSpacing" : "0em",
"HeadingExtraLargeLineHeight" : "1.25",
"HeadingExtraLargeTextDecoration" : "none",
"HeadingExtraLargeTextTransform" : "none",
"HeadingLargeColor" : "var(--dxp-g-root-contrast)",
"HeadingLargeFont" : "Salesforce Sans",
"HeadingLargeFontSize" : "1.75rem",
"HeadingLargeFontStyle" : "normal",
"HeadingLargeFontWeight" : "300",
"HeadingLargeLetterSpacing" : "0em",
"HeadingLargeLineHeight" : "1.25",
"HeadingLargeTextDecoration" : "none",
"HeadingLargeTextTransform" : "none",
"HeadingMediumColor" : "var(--dxp-g-root-contrast)",
"HeadingMediumFont" : "Salesforce Sans",
"HeadingMediumFontSize" : "1.25rem",
"HeadingMediumFontStyle" : "normal",
"HeadingMediumFontWeight" : "300",
"HeadingMediumLetterSpacing" : "0em",
"HeadingMediumLineHeight" : "1.25",
"HeadingMediumTextDecoration" : "none",
"HeadingMediumTextTransform" : "none",
"HeadingSmallColor" : "var(--dxp-g-root-contrast)",
"HeadingSmallFont" : "Salesforce Sans",
"HeadingSmallFontSize" : "1.125rem",
"HeadingSmallFontStyle" : "normal",
"HeadingSmallFontWeight" : "300",
"HeadingSmallLetterSpacing" : "0em",
"HeadingSmallLineHeight" : "1.25",
"HeadingSmallTextDecoration" : "none",
"HeadingSmallTextTransform" : "none",
"HorizontalRowPaddingDesktop" : "1rem",
"HorizontalRowPaddingMobile" : "0.75rem",
"LinkColor" : "var(--dxp-g-brand)",
"LinkHoverColor" : "var(--dxp-s-link-text-color-1)",
```

```
"LinkTextDecoration" : "none",
"LinkTextDecorationFocus" : "underline",
"LinkTextDecorationHover" : "underline",
"MaxContentWidthDesktop" : "1800px",
"MaxContentWidthMobile" : "none",
"MobileBaseFontSize" : "1rem",
"PrimaryAccentColor" : "#005fb2",
"PrimaryAccentForegroundColor" : "#ffffff",
"SiteLogo" : "",
"TextColor" : "#1a1b1e",
"VerticalRowPaddingDesktop" : "1rem",
"VerticalRowPaddingMobile" : "0.75rem",
"_BackgroundColor1" : "#ebebeb",
"_BackgroundColor2" : "#c2c2c2",
"_BackgroundColor3" : "#858585",
"_ButtonActiveColorContrast" : "var(--dxp-g-brand-contrast-1)",
"_ButtonColor1" : "var(--dxp-g-brand-1)",
"_ButtonColorContrast" : "var(--dxp-g-brand-contrast)",
"_ButtonFocusColorContrast" : "var(--dxp-g-brand-contrast-1)",
"_ButtonHoverColorContrast" : "var(--dxp-g-brand-contrast-1)",
"_DestructiveColor" : "#c23934",
"_DestructiveColor1" : "#a2302b",
"_DestructiveColor2" : "#611d1a",
"_DestructiveColor3" : "#010000",
"_DestructiveForegroundColor" : "#ffffff",
"_DestructiveForegroundColor1" : "#ffffff",
"_DestructiveForegroundColor2" : "#ffffff",
"_DestructiveForegroundColor3" : "#ffffff",
"_InfoColor" : "#16325c",
"_InfoColor1" : "#0e203b",
"_InfoColor2" : "#000000",
"_InfoColor3" : "#000000",
"_InfoForegroundColor" : "#ffffff",
"_InfoForegroundColor1" : "#ffffff",
"_InfoForegroundColor2" : "#ffffff",
"_InfoForegroundColor3" : "#ffffff",
"_LinkColor1" : "var(--dxp-g-brand-1)",
"_NeutralColor" : "#ecebea",
"_NeutralColor1" : "#d9d7d5",
"_NeutralColor2" : "#b2aeaa",
"_NeutralColor3" : "#76716b",
"_NeutralForegroundColor" : "#000000",
"_NeutralForegroundColor1" : "#000000",
"_NeutralForegroundColor2" : "#000000",
"_NeutralForegroundColor3" : "#ffffff",
"_OfflineColor" : "#444444",
"_OfflineColor1" : "#303030",
"_OfflineColor2" : "#070707",
"_OfflineColor3" : "#000000",
"_OfflineForegroundColor" : "#ffffff",
"_OfflineForegroundColor1" : "#ffffff",
"_OfflineForegroundColor2" : "#ffffff",
"_OfflineForegroundColor3" : "#ffffff",
"_PrimaryAccentColor1" : "#004989",
```

```
      "_PrimaryAccentColor2" : "#001e38",
      "_PrimaryAccentColor3" : "#000000",
      "_PrimaryAccentForegroundColor1" : "#ffffff",
      "_PrimaryAccentForegroundColor2" : "#ffffff",
      "_PrimaryAccentForegroundColor3" : "#ffffff",
      "_SiteLogoUrl" : "",
      "_SuccessColor" : "#4bca81",
      "_SuccessColor1" : "#36b66c",
      "_SuccessColor2" : "#237747",
      "_SuccessColor3" : "#07190f",
      "_SuccessForegroundColor" : "#000000",
      "_SuccessForegroundColor1" : "#000000",
      "_SuccessForegroundColor2" : "#ffffff",
      "_SuccessForegroundColor3" : "#ffffff",
      "_TextColor1" : "#000000",
      "_TextColor2" : "#000000",
      "_TextColor3" : "#000000",
      "_WarningColor" : "#ffb75d",
      "_WarningColor1" : "#ffa534",
      "_WarningColor2" : "#e27d00",
      "_WarningColor3" : "#673900",
      "_WarningForegroundColor" : "#000000",
      "_WarningForegroundColor1" : "#000000",
      "_WarningForegroundColor2" : "#000000",
      "_WarningForegroundColor3" : "#ffffff"
    }
  }
}
```

sfdc_cms__languageSettings Folder

This content type folder contains one content subfolder called languages. The languages content subfolder contains two or more `JSON` files:

- `_meta.json`
- `content.json`
- If applicable, additional `JSON` files that represent variations of the content item

<apiName>`/content.json`

Property	Description
`defaultLocale`	**Field Type** string **Description** Represents the default locale for a workspace. The language listed here is the default fallback language for all of the workspace's languageTypes.
`languages`	**Field Type** languageType[]

Property	Description
	Description Required. Represents the languages available in the space. Each `languageType` contains values related to that particular language.
`languageType.fallbacklocale`	**Field Type** string **Description** Represents the locale to use when no content is available for the selected locale. For example, if a site visitor chooses Japanese from the language selector but there's no content for that page in Japanese, content is displayed in the fallback locale. The fallbackLocale's `isAuthoringOnly` value must be `false`. Only one level of fallback is allowed. Here are examples for a workspace with a locale of `en_US`, and with Spanish, French, and Finnish as supported locales. • Not allowed: Spanish falls back to French, and French falls back to Finnish. This configuration includes two levels of fallback. • Allowed: Spanish falls back to French, and French falls back to English. `en_US` is the workspace's default. • Allowed: Spanish falls back to French, and French has no fallback. This property is displayed only if the `languageType.fallbackLocale` is different from the workspace's default locale.
`languageType.isActive`	**Field Type** boolean **Description** Indicates whether the language is active (`true`) or inactive (`false`). The only supported value is `true`.
`languageType.isAuthoringOnly`	**Field Type** boolean **Description** Indicates whether the language is scoped only to authoring workspace (`true`) or not (`false`). The default value is `false`.

Property	Description
	When set to `true`, the `languageType` and its associated translations aren't visible to site visitors. When set to `false`, site visitors can see the `languageType` and its associated translations.
`languageType.label`	**Field Type** string **Description** Required. Defines the display label for a language. The display label appears in any language selector components that you add to your site and in the language selector in Experience Builder. The default value is the locale name. For example, if `languageType.locale=fr`, then `languageType.label=French`.
`languageType.locale`	**Field Type** string **Description** Required. Represents the languageCode appended with the countryCode, separated by an underscore. For example, `en_US` indicates a languageCode of English, and a countryCode of United States. Defines the locale, or geographic region, of a language. Locales determine the display format for date and time, user names, address, and commas and periods in numbers. Each `languageType.locale` and `languageType.label` combination can be used only one time per workspace.
`title`	**Field Type** string **Description** Required. Represents the label of the content item.
`type`	**Field Type** string **Description** Required.

Property	Description
	Represents the content type. The only allowed value is `sfdc_cms__languageSettings`.

```
{
  "type" : "sfdc_cms__languageSettings",
  "title" : "LanguageContent",
  "contentBody" : {
    "languages" : [ {
      "locale" : "en_US",
      "label" : "English (US)",
      "isActive" : true,
      "isAuthoringOnly" : false
    } ],
    "defaultLocale" : "en_US"
  }
}
```

sfdc_cms__route Folder

This content type folder contains one content subfolder for each of the site's routes. Each route content subfolder contains two or more `JSON` files:

- `_meta.json`
- `content.json`
- If applicable, additional `JSON` files that represent variations of the content item

<apiName>`/content.json`

Property	Description
`activeViewId`	**Field Type** content link **Description** Required. Represents the `apiName` of the route's active view. Must reference an existing view in the same workspace.
`configurationTags`	**Field Type** string[] **Description** Represents the configuration tags for the route. The only allowed `value is all-in-static-site`. Note: Don't edit this property, because it's an internal property.

Property	Description
`pageAccess`	**Field Type** enumeration of type string **Description** Identifies the status of a route as public or private. When set to the default value `UseParent`, the status of the site determines the status of the route. This value isn't editable from the user interface for routes that are always private. Valid values are: • `UseParent` • `Public` • `RequiresLogin` Note: You can change the `pageAccess` value only if the site is authenticated.
`routeType`	**Field Type** string **Description** Required. Identifies the type of route. The `routeType` value must be unique within a workspace. This value must match the `viewType on page 853` value in the corresponding view content item.
`title`	**Field Type** string **Description** Required. Represents the label of the route.
`type`	**Field Type** string **Description** Required. Represents the content type. The only supported value is `sfdc_cms__route`.
`urlPrefix`	**Field Type** string **Description** Required.

Property	Description
	Represents the base URL for the route.

```
{
  "type" : "sfdc_cms__route",
  "title" : "Error",
  "contentBody" : {
    "activeViewId" : "error",
    "configurationTags" : [ ],
    "pageAccess" : "UseParent",
    "routeType" : "error",
    "urlPrefix" : "error"
  }
}
```

sfdc_cms__site Folder

This content type folder exists at the root level and contains one content subfolder. The content subfolder contains two or more `JSON` files:

- `_meta.json`
- `content.json`
- If applicable, additional `JSON` files that represent variations of the content item

<apiName>`/content.json`

Property	Description
`authenticationType`	**Field Type** enumeration of type string **Description** Required. Indicates whether guest users have access to the site. Valid values are: • `AUTHENTICATED`—The site isn't public. Only authenticated users can access the site after logging in. • `AUTHENTICATED_WITH_PUBLIC_ACCESS_ENABLED`—The site is an authenticated site, but the **Public can access the site** checkbox is enabled in Experience Builder in **Settings** > **General**. Guest users can access the site. • `UNAUTHENTICATED`—The unauthenticated site is publicly available to anyone on the web, and doesn't support login or authentication. `UNAUTHENTICATED` is a legacy value. We recommend using `AUTHENTICATED_WITH_PUBLIC_ACCESS_ENABLED` access to allow guest user access.

Property	Description
`title`	**Field Type** string **Description** Required. Represents the label of the content item.
`type`	**Field Type** string **Description** Required. Represents the content type. The only allowed value is `sfdc_cms__site`.

```
{
  "type" : "sfdc_cms__site",
  "title" : "Capricorn_Coffee",
  "contentBody" : {
    "authenticationType" : "AUTHENTICATED"
  }
}
```

sfdc_cms__theme Folder

This content type folder contains one content subfolder, representing the site's theme. The content subfolder contains two or more `JSON` files:

- `_meta.json`
- `content.json`
- If applicable, additional `JSON` files that represent variations of the content item

<apiName>`/content.json`

Property	Description
`activeBrandingSetId`	**Field Type** content link **Description** Required. Represents the `apiName` of the site's active branding set. Must reference an existing brandingSet with a type of `APP` in the same workspace.

Property	Description
`definitionName`	**Field Type** string **Description** Required. Represents the definition name of a theme. Must match the template's theme definition name.
`layouts`	**Field Type** layout[] **Description** Required. Represents an array of the theme layouts available in the workspace and provides details about each theme layout. The template and features used in the workspace determine which layouts are required. For example, the `ServiceNotAvailable` layout is required if the site includes a Service Not Available page.
`layouts.layoutId`	**Field Type** content link **Description** Required. Represents the `apiName` of an available theme layout. Must refer to an existing theme layout in the same workspace.
`layouts.layoutType`	**Field Type** string **Description** Represents the name of the theme layout specified in Experience Builder. An inner theme layout is required. The inner theme layout is applied by default to all of a site's pages except for login pages.
`title`	**Field Type** string **Description** Required. Represents the name of the theme.
`type`	**Field Type** string

Property	Description
	Description Required. Represents the content type. The only allowed value is `sfdc_cms__theme`.

```
{
  "type" : "sfdc_cms__theme",
  "title" : "Build Your Own (LWR)",
  "contentBody" : {
    "activeBrandingSetId" : "Build_Your_Own_LWR",
    "definitionName" : "byo",
    "layouts" : [ {
      "layoutId" : "snaThemeLayout",
      "layoutType" : "ServiceNotAvailable"
    }, {
      "layoutId" : "scopedHeaderAndFooter",
      "layoutType" : "Inner"
    } ]
  }
}
```

sfdc_cms__themeLayout Folder

This content type folder contains one content subfolder for each theme layout in the site. Each content subfolder contains two or more `JSON` files:

- `_meta.json`
- `content.json`
- If applicable, additional `JSON` files that represent variations of the content item

Note: We recommend that you don't add, reorder, or delete a component within a locked region using the DigitalExperienceBundle. To find out which regions are locked, in Experience Builder, view the Page Structure tab for the page that you're working on. If the region that you're modifying has a lock icon next to it, it's a locked region.

<apiName>`/content.json`

Property	Description
`children`	**Field Type** regionType[] **Description** Represents individual regions within a layout.
`component`	**Field Type** componentType

Property	Description
	Description Required. Contains the layout details for this theme layout in regions and components.
`componentType.attributes`	**Field Type** map **Description** A map of design attribute values used by the component definition to render the definition.
`componentType.attributes.dxpStyle`	**Field Type** map **Description** Lets you define visibility, padding, and margin values for a component. Visibility controls whether the component is visible on desktop, tablet, and mobile versions of your site. Margin and padding are map types that contain string properties for top, right, bottom, and left spacing. Padding controls space within a component. Margin controls space around the outside of a component. Units for margin and padding can be px, rem, or %, and all four spacing values must use the same unit. This field is available in API version 57.0 and later.
`componentType.children`	**Field Type** regionType[] **Description** Represents regions within a component.
`componentType.definition`	**Field Type** string **Description** Required. Represents the name of the Lightning Web Component (LWC) using the format namespace:componentName. Must be one of the theme layouts defined as part of the default theme, or use the target lightningCommunity__Theme_Layout. The default theme layouts differ per theme. For example, the Build Your Own (BYO) template uses the BYO theme, with the allowed theme

Property	Description
	layouts `community_byo:scopedHeaderandFooter` and `community_layout:simpleThemeLayout`.
`componentType.id`	**Field Type** UUID **Description** Required. Represents the UUID of a component.
`componentType.scopedBrandingSetId`	**Field Type** content link **Description** Represents the brandingSet that applies to this component. The brandingSet must be of type `SCOPED`. This property is applicable only if the component's `definitionName` is `community_layout:section`.
`componentType.customCssClasses`	**Field Type** string **Description** Represents the custom CSS classes applied to the host element of the component. Must be a space-delimited list of valid CSS classes. This field is available in API version 59.0 and later.
`componentType.title`	**Field Type** string **Description** Represents a component variation name. Available in API version 59.0 and later.
`componentType.type`	**Field Type** string **Description** Required. Allowed values are: • `component` • `mutationComponent`
`componentType.variations`	**Field Type** componentType[]

Property	Description
	Description Represents a component variation within a component. Available in API version 59.0 and later.
`componentVariationType.rule`	**Field Type** rule type **Description** A visibility rule associated with a component variation. Available in API version 59.0 and later.
`componentVariationType.variationId`	**Field Type** string **Description** The unique ID of a component variation. Available in API version 59.0 and later.
`contentOperations`	**Field Type** contentOperationType **Description** A content operation of the variation or visibility type.
`contentOperationsType.operations`	**Field Type** operationsType[] **Description** Component visibility operation.
`expressionCriteriaType.criterionNumber`	**Field Type** integer **Description** The sequence number of an expression criteria. Available in API version 59.0 and later.
`expressionCriteriaType.operator`	**Field Type** string **Description** Operator of the criteria or condition in a visibility rule. Supported operators are: • `EQUAL` • `NE`: Does not equal • `CONTAINS` • `NC`: Does not contain • `LT`: Less than

Property	Description
	• `GT`: Greater than • `LE`: Less than or equal to • `GE`: Greater than or equal to • `StartsWith`
`expressionCriteriaType.resource`	**Field Type** string **Description** The resource of the criteria or condition in a visibility rule.
`expressionCriteriaType.value`	**Field Type** object **Description** The value of the criteria or condition in a visibility rule. The value can be number, boolean, or string.
`operationsType.isActive`	**Field Type** boolean **Description** When `true`, the visibility operation is active.
`operationsType.isHiddenOnOperationSuccess`	**Field Type** boolean **Description** When `true`, the component is hidden when the operation is successful.
`operationsType.rule`	**Field Type** RuleType **Description** The component visibility rule.
`operationsType.ruleToVariationList`	**Field Type** componentVariationType[] **Description** Variation rules associated with a specific component ID. Only 15 variations are allowed in one ruleToVariationList, which means only 15 variations can correspond to a specific targetId. Available in API version 59.0 and later.
`operationsType.targetId`	**Field Type** string

Property	Description
	Description The component ID where the visibility rule is applied.
`regionType.children`	**Field Type** componentType[] **Description** Represents individual components within the region.
`regionType.id`	**Field Type** UUID **Description** Required. Represents the UUID of a region.
`regionType.name`	**Field Type** string **Description** Required. Represents the name of the region.
`regionType.title`	**Field Type** string **Description** Represents the label for a region.
`regionType.type`	**Field Type** string **Description** Required. `region` is the only allowed value.
`regionType.variations`	**Field Type** componentType[] **Description** Represents a component variation within a region. Available in API version 59.0 and later.
`ruleType.criteriaType`	**Field Type** string **Description** The condition or criteria type of the rule. Allowed values include `AllCriteriaMatch` and `AnyCriterionMatches`.

Property	Description
`ruleType.customFormula`	**Field Type** string **Description** A custom formula created from expression criteria for a visibility rule. Available in API version 59.0 and later.
`ruleType.description`	**Field Type** string **Description** The description of a visibility rule.
`ruleType.expressionCriteria`	**Field Type** expressionCriteriaType[] **Description** The expression criteria of a visibility rule.
`ruleType.name`	**Field Type** string **Description** The name of a visibility rule.
`title`	**Field Type** string **Description** Required. Represents the label of the content item.
`type`	**Field Type** string **Description** Required. Represents the content type. The only allowed value is `sfdc_cms__themeLayout`.

```
{
  "type" : "sfdc_cms__themeLayout",
  "title" : "Service Not Available Theme Layout",
  "contentBody" : {
    "component" : {
      "id" : "50458146-12d8-4dd7-a37b-62f71615c1a0",
      "type" : "component",
      "children" : [ {
```

```
        "title": "Theme Header",
          "id": "8fc2497b-fae7-4570-bc69-63a7a229e6fc",
          "type": "region",
          "name": "header",
          "children": [
            {
              "id": "05224a3a-8044-9h2h-9187-b3f43155344d",
              "type": "component",
              "definition": "community_builder:htmlEditor",
              "attributes": {
                "richTextValue": "<div style=\"display: flex; justify-content: center;
align-items: center; margin: 50px 0; flex-direction: column; text-align: center;\"><div
style=\"background-image: url(assets/img/desert.svg); background-size: cover;
background-position: center; height: 300px; width: 100%; max-width: 600px; min-width:
300px;\"></div><h1 class=\"slds-text-heading_medium slds-p-bottom_x-small\">Start Building
 Your Page</h1> <div>Drag and drop a component into the content slots.</div></div>"
              },
              "variations": [
                {
                  "id": "5d700461-4c2a-4930-98e5-366ec2a4d41e",
                  "title": "Variation Country USA",
                  "type": "mutationComponent",
                  "definition": "community_builder:htmlEditor",
                  "attributes": {
                   "richTextValue": "<div style=\"display: flex; justify-content: center;
 align-items: center; margin: 50px 0; flex-direction: column; text-align: center;\"><div
style=\"background-image: url(assets/img/desert.svg); background-size: cover;
background-position: center; height: 300px; width: 100%; max-width: 600px; min-width:
300px;\"></div><h1 class=\"slds-text-heading_medium slds-p-bottom_x-small\">Start Building
 Your Page</h1> <div>Drag and drop a component into the content slots.</div></div>"
                  }
                }
              ]
            }
          ]
        },
        {
        "title" : "Theme Footer",
        "id" : "05224a3a-8044-9h2h-9187-b3f43155344d",
        "type" : "region",
        "name" : "footer"
      } ],
      "definition" : "community_layout:simpleThemeLayout",
      "attributes": {}
    },    "contentOperations": {
      "operations": [
        {
          "targetId": "05224a3a-8044-9h2h-9187-b3f43155344d",
          "isHiddenOnOperationSuccess": false,
          "isActive": true,
          "rule": {
            "name": "708b6ff0-d50c-4cea-a492-kkjk4b174e86",
            "description": "",
            "criteriaType": "AllCriteriaMatch",
```

```
            "expressionCriteria": [
              {
                "resource": "User.Record.FirstName",
                "operator": "Equal",
                "value": "Clark"
              }
            ]
          }
        },
        {
          "targetId": "05224a3a-8044-9h2h-9187-b3f43155344d",
          "ruleToVariationList": [
            {
              "variationId": "5d700461-4c2a-4930-98e5-366ec2a4d41e",
              "rule": {
                "name": "6e1ea488-13d8-477a-a9d7-93f442cc7936",
                "description": "",
                "criteriaType": "CustomLogicMatches",
                "customFormula": "(1 && 3) || 2",
                "expressionCriteria": [
                  {
                    "resource": "User.Record.Country",
                    "operator": "Equal",
                    "value": "USA",
                    "criterionNumber": 1
                  },
                  {
                    "resource": "User.Record.City",
                    "operator": "Equal",
                    "value": "Chicago",
                    "criterionNumber": 2
                  },
                  {
                    "resource": "User.Record.PostalCode",
                    "operator": "Equal",
                    "value": "60131",
                    "criterionNumber": 3
                  }
                ]
              }
            }
          ]
        }
      ]
    }
  }
}
```

sfdc_cms__view Folder

This content type folder contains one content subfolder per view. Each content subfolder contains two or more `JSON` files:

- `_meta.json`
- `content.json`

- If applicable, additional `JSON` files that represent variations of the content item

Each Experience Builder site is built from single-page applications, which are web apps that load a single HTML page. Single-page applications consist of multiple views that update the page dynamically as the user interacts with it. A view is made up of regions that contain other regions or components in the rendered page for the user. Single-page applications in your site are defined in the `sfdc_cms__appPage` folder.

Each `content.json` file in the sfdc_cms__view folder contains a hidden region named `sfdcHiddenRegion`. The hidden region contains a component with a definition of `community_builder:seoAssistant` that represents the SEO assistant component. This component corresponds to the SEO page properties that you can configure in Experience Builder and isn't visible on your pages. To improve search engine results, use the SEO assistant component to set the `customHeadTags`, `description`, and `pageTitle` properties for your public and custom site pages. You can't edit the other properties associated with the SEO assistant component. To learn more about what the title, description, and custom head tags properties represent and which head tags are allowed, see SEO Page Properties in Experience Builder.

Note: We recommend that you don't add, reorder, or delete a component within a locked region using the DigitalExperienceBundle. To find out which regions are locked, in Experience Builder, view the Page Structure tab for the page that you're working on. If the region that you're modifying has a lock icon next to it, it's a locked region.

`<apiName>/content.json`

Property	Description
`children`	**Field Type** regionType[] **Description** Represents individual regions within a view.
`component`	**Field Type** componentType **Description** Required. Represents the view layout and contains regions and components that help render a view.
`componentType.attributes`	**Field Type** map **Description** A map of design attribute values used by the component definition to render the definition.
`componentType.attributes.dxpStyle`	**Field Type** map **Description** Lets you define visibility, padding, and margin values for a component. Visibility controls whether the component is visible on desktop, tablet, and mobile versions of your site.

Property	Description
	Margin and padding are map types that contain string properties for top, right, bottom, and left spacing. Padding controls space within a component. Margin controls space around the outside of a component. Units for margin and padding can be px, rem, or %, and all four spacing values must use the same unit. This field is available in API version 57.0 and later.
`componentType.children`	**Field Type** regionType[] **Description** Represents regions within a component.
`componentType.definition`	**Field Type** string **Description** Required. Represents the name of the Lightning Web Component (LWC) in the format namespace:componentName.
`componentType.id`	**Field Type** UUID **Description** Required. Represents the UUID of a component.
`componentType.scopedBrandingSetId`	**Field Type** content link **Description** Represents the brandingSet that applies to this component. The brandingSet must be of type `SCOPED`. This property is applicable only if the component's `definitionName` is `community_layout:section`.
`componentType.customCssClasses`	**Field Type** string **Description** Represents the custom CSS classes applied to the host element of the component. Must be a space-delimited list of valid CSS classes. This field is available in API version 59.0 and later.
`componentType.title`	**Field Type** string

Property	Description
	Description Represents a component variation name. Available in API version 59.0 and later.
`componentType.type`	**Field Type** string **Description** Required. Allowed values are: • `component` • `mutationComponent`
`componentType.variations`	**Field Type** componentType[] **Description** Represents a component variation within a component. Available in API version 59.0 and later.
`componentVariationType.rule`	**Field Type** rule type **Description** A visibility rule associated with a component variation. Available in API version 59.0 and later.
`componentVariationType.variationId`	**Field Type** string **Description** The unique ID of a component variation. Available in API version 59.0 and later.
`contentOperations`	**Field Type** contentOperationType **Description** A content operation of the variation or visibility type.
`expressionCriteriaType.criterionNumber`	**Field Type** integer **Description** The sequence number of an expression criteria. Available in API version 59.0 and later.

Property	Description
`contentOperationsType.operations`	**Field Type** operationsType[] **Description** Component visibility operation.
`expressionCriteriaType.operator`	**Field Type** string **Description** Operator of the criteria or condition in a visibility rule. Supported operators are: • `EQUAL` • `NE`: Does not equal • `CONTAINS` • `NC`: Does not contain • `LT`: Less than • `GT`: Greater than • `LE`: Less than or equal to • `GE`: Greater than or equal to • `StartsWith`
`expressionCriteriaType.resource`	**Field Type** string **Description** The resource of the criteria or condition in a visibility rule.
`expressionCriteriaType.value`	**Field Type** object **Description** The value of the criteria or condition in a visibility rule. The value can be number, boolean, or string.
`operationsType.isActive`	**Field Type** boolean **Description** When `true`, the visibility operation is active.
`operationsType.isHiddenOnOperationSuccess`	**Field Type** boolean **Description** When `true`, the component is hidden when the operation is successful.

Property	Description
`operationsType.rule`	**Field Type** RuleType **Description** The component visibility rule.
`operationsType.ruleToVariationList`	**Field Type** componentVariationType[] **Description** Variation rules associated with a specific component ID. Only 15 variations are allowed in one ruleToVariationList, which means only 15 variations can correspond to a specific targetId. Available in API version 59.0 and later.
`operationsType.targetId`	**Field Type** string **Description** The component ID where the visibility rule is applied.
`regionType.children`	**Field Type** componentType[] **Description** Represents individual components within the region.
`regionType.id`	**Field Type** UUID **Description** Required. Represents the UUID of a region.
`regionType.name`	**Field Type** string **Description** Required. Represents the name of the region.
`regionType.title`	**Field Type** string **Description** Represents the label for a region.
`regionType.type`	**Field Type** string

Property	Description
	Description Required. `region` is the only allowed value.
`regionType.variations`	**Field Type** componentType[] **Description** Represents a component variation within a region. Available in API version 59.0 and later.
`ruleType.criteriaType`	**Field Type** string **Description** The condition or criteria type of the rule. Allowed values include `AllCriteriaMatch` and `AnyCriterionMatches`.
`ruleType.customFormula`	**Field Type** string **Description** A custom formula created from expression criteria for a visibility rule. Available in API version 59.0 and later.
`ruleType.description`	**Field Type** string **Description** The description of a visibility rule.
`ruleType.expressionCriteria`	**Field Type** expressionCriteriaType[] **Description** The expression criteria of a visibility rule.
`ruleType.name`	**Field Type** string **Description** The name of a visibility rule.
`themeLayoutType`	**Field Type** string **Description** Required. Represents the `apiName` of the view's active theme layout.

Property	Description
	Must reference an existing theme layout in the same workspace.
`title`	**Field Type** string **Description** Represents the name of the view. Limited to 80 characters. Note: We recommend keeping the `title` value and the `pageTitle` value in sync. If you update either value for a view, ensure that you manually update the other value in the same `content.json` file to match.
`type`	**Field Type** string **Description** Required. Represents the content type. The only allowed value is `sfdc_cms__view`.
`viewType`	**Field Type** string **Description** Required. Identifies the type of view. The `viewType` value must be unique within a workspace. This value must match the `routeType on page 840` value in the corresponding route content item.

```
{
  "type" : "sfdc_cms__view",
  "title" : "Home",
  "contentBody" : {
    "themeLayoutType" : "Inner",
    "viewType" : "home",
    "component" : {
      "id" : "40c14c97-1846-4872-8e9e-cdf3d11beb34",
      "type" : "component",
      "children" : [ {
        "title" : "Content",
        "id" : "c507f23d-6e2a-457a-9656-2377846dd639",
        "type" : "region",
        "children" : [ {
          "id" : "21f99012-3a2f-488e-bf48-f782dc7b4300",
```

```
          "type" : "component",
          "children" : [ {
            "title" : "Column 1",
             "id": "05224a3a-8044-4bfb-9187-b3f43155344d",
            "type" : "region",
            "children" : [ {
              "id" : "05224a3a-8044-4bfb-9187-b3f43155344d",
              "type" : "component",
              "definition" : "community_builder:htmlEditor",
              "attributes" : {
                "dxpStyle" : {
                  "isVisible" : false,
                  "margin" : {
                    "bottom" : "20px",
                    "left" : "20px",
                    "right" : "20px",
                    "top" : "20px"
                  },
                  "padding" : {
                    "bottom" : "20px",
                    "left" : "20px",
                    "right" : "20px",
                    "top" : "20px"
                  }
                },
                "richTextValue" : "<div style=\"display: flex; justify-content: center;
align-items: center; margin: 50px 0; flex-direction: column; text-align: center;\"><div
style=\"background-image: url(assets/img/desert.svg); background-size: cover;
background-position: center; height: 300px; width: 100%; max-width: 600px; min-width:
300px;\"></div><h1 class=\"slds-text-heading_medium slds-p-bottom_x-small\">Start Building
 Your Page</h1> <div>Drag and drop a component into the content slots.</div></div>"
              },
               "variations": [
                        {
                          "id": "5d700461-4c2a-4930-98e5-366ec2a4d41e",
                          "title": "Variation Country USA",
                          "type": "mutationComponent",
                          "definition": "community_builder:htmlEditor",
                          "attributes": {
                           "richTextValue": "<div style=\"display: flex; justify-content:
 center; align-items: center; margin: 50px 0; flex-direction: column; text-align:
center;\"><div style=\"background-image: url(assets/img/desert.svg); background-size:
cover; background-position: center; height: 300px; width: 100%; max-width: 600px; min-width:
 300px;\"></div><h1 class=\"slds-text-heading_medium slds-p-bottom_x-small\">Start Building
 Your Page</h1> <div>Drag and drop a component into the content slots.</div></div>"
                          }
                        }
                      ]
              "customCssClasses": "myClass"
            } ],
            "name" : "col1"
          } ],
          "definition" : "community_layout:section",
          "attributes" : {
```

```
            "backgroundImageConfig" : "",
            "backgroundImageOverlay" : "rgba(0,0,0,0)",
            "sectionConfig" :
"{\"UUID\":\"21f99012-3a2f-488e-bf48-f782dc7b4300\",\"columns\":[{\"UUID\":\"5019aeab-6437-4194-8369-22c19aa45dc9\",\"columnName\":\"Colum
 1\",\"columnKey\":\"col1\",\"columnWidth\":\"12\",\"seedComponents\":null}]}"
          }
        } ],
        "name" : "content"
      }, {
        "title" : "sfdcHiddenRegion",
        "id" : "8157e041-9c41-460a-b596-c45babbbd53b",
        "type" : "region",
        "children" : [ {
          "id" : "2d536aae-a859-4264-ba9e-9a569daf7213",
          "type" : "component",
          "definition" : "community_builder:seoAssistant",
          "attributes" : {
            "customHeadTags" : "",
            "description" : "",
            "pageTitle" : "Home",
            "recordId" : "{!recordId}"
          }
        } ],
        "name" : "sfdcHiddenRegion"
      } ],
      "definition": "community_layout:sldsFlexibleLayout"
    },
    "contentOperations": {
      "operations": [
        {
          "targetId": "05224a3a-8044-4bfb-9187-b3f43155344d",
          "isHiddenOnOperationSuccess": false,
          "isActive": true,
          "rule": {
            "name": "a53bb452-003f-4015-a751-0403c70731a1",
            "description": "",
            "criteriaType": "AllCriteriaMatch",
            "expressionCriteria": [
              {
                "resource": "User.isGuest",
                "operator": "Equal",
                "value": false
              }
            ]
          }
        },
        {
          "targetId": "05224a3a-8044-9h2h-9187-b3f43155344d",
          "ruleToVariationList": [
            {
              "variationId": "5d700461-4c2a-4930-98e5-366ec2a4d41e",
              "rule": {
                "name": "6e1ea488-13d8-477a-a9d7-93f442cc7936",
                "description": "",
```

```
                "criteriaType": "CustomLogicMatches",
                "customFormula": "(1 && 3) || 2",
                "expressionCriteria": [
                  {
                    "resource": "User.Record.Country",
                    "operator": "Equal",
                    "value": "USA",
                    "criterionNumber": 1
                  },
                  {
                    "resource": "User.Record.City",
                    "operator": "Equal",
                    "value": "Chicago",
                    "criterionNumber": 2
                  },
                  {
                    "resource": "User.Record.PostalCode",
                    "operator": "Equal",
                    "value": "60131",
                    "criterionNumber": 3
                  }
                ]
              }
            }
          ]
        }
      ]
    }
  }
}
```

DigitalExperienceConfig

Represents details for your organization's workspaces, such as the site label, site URL path prefix, and workspace type.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

DigitalExperienceConfig components have the suffix `.digitalExperienceConfig` and are stored in the `digitalExperienceConfigs` folder.

Version

DigitalExperienceConfig components are available in API version 56.0 and later.

Special Access Rules

You can use DigitalExperienceConfig for enhanced LWR sites created after the Winter '23 release.

Fields

Field Name	Description
`label`	**Field Type** string **Description** Required. The name of the site.
`site`	**Field Type** Site **Description** Required. Contains site-related settings, such as the site's URL path prefix.
`space`	**Field Type** string **Description** Required. References the workspace that contains the site's content items such as brandingSets, themes, views, and routes.

Site

Represents site-related information, such as the URL path prefix.

Field Name	Description
`urlPathPrefix`	**Field Type** string **Description** The first part of the path on the site's URL that distinguishes this site from other sites. For example, if your site URL is *`MyDomainName`*.my.site.com/partners, then partners is the `urlPathPrefix`.

Declarative Metadata Sample Definition

The following is an example of a DigitalExperienceConfig component.

```
<?xml version="1.0" encoding="UTF-8"?>
<DigitalExperienceConfig xmlns="http://soap.sforce.com/2006/04/metadata">
    <label>Capricorn_Coffee</label>
    <site>
        <urlPathPrefix>CapricornCoffee</urlPathPrefix>
    </site>
    <space>site/Capricorn_Coffee1</space>
</DigitalExperienceConfig>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>Capricorn_Coffee1</members>
        <name>DigitalExperienceConfig</name>
    </types>
    <version>56.0</version>
</Package>
```

Usage

To retrieve and deploy DigitalExperienceConfig, use legacy sfdx commands.

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

DisclosureDefinition

Represents information that defines a disclosure type, such as details of the publisher or vendor who created or implemented the report.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

DisclosureDefinition components have the suffix `.disclosureDefinition` and are stored in the `disclosureDefinitions` folder.

Version

DisclosureDefinition components are available in API version 57.0 and later.

Special Access Rules

The DisclosureAndComplianceHubAddOn license is required to access this object along with user access for the Disclosure Compliance Hub permission set license.

Fields

Field Name	Description
`description`	**Field Type** string **Description** The description about the disclosure definition.
`disclosureType`	**Field Type** string **Description** Required. The API name of the disclosure type associated with this definition.
`isProtected`	**Field Type** boolean **Description** An auto-generated value that doesn't impact the behavior of the metadata type. The default is `false`.
`masterLabel`	**Field Type** string **Description** Required. A user-friendly name for DisclosureDefinition, which is defined when the DisclosureDefinition is created.

Declarative Metadata Sample Definition

The following is an example of a DisclosureDefinition component.

```
<?xml version="1.0" encoding="UTF-8"?>
<DisclosureDefinition
 xmlns="http://soap.sforce.com/2006/04/metadata">
 <description>This is GRI Disclosure Definition</description>
 <disclosureType>disclstype10</disclosureType>
 <isProtected>false</isProtected>
```

```
 <masterLabel>GRI</masterLabel>
</DisclosureDefinition>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package
 xmlns="http://soap.sforce.com/2006/04/metadata">
 <fullName>Pkg</fullName>
 <types>
  <members>GRI</members>
  <name>DisclosureDefinition</name>
 </types>
 <types>
  <members>dt12</members>
  <name>DisclosureType</name>
 </types>
 <version>57.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

DisclosureDefinitionVersion

Represents the version information about the disclosure definition.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

DisclosureDefinitionVersion components have the suffix `.disclosureDefinitionVersion` and are stored in the `disclosureDefinitionVersions` folder.

Version

DisclosureDefinitionVersion components are available in API version 57.0 and later.

Special Access Rules

The DisclosureAndComplianceHubAddOn and OmniStudioDesignerAddon licenses are required to access this object along with user access for the Disclosure Compliance Hub and OmniStudio Admin permission set licenses.

Fields

Field Name	Description
`authoringMode`	**Field Type** AuthoringMode (enumeration of type string) **Description** Specifies the authoring mode used to launch the disclosure authoring experience. Possible values are: • `Microsoft 365 Word` • `Omniscript and Microsoft 365 Word` • `Omniscript Form`
`description`	**Field Type** string **Description** The description about the disclosure definition version.
`disclosureDefCurrVer`	**Field Type** string **Description** For internal use only.
`disclosureDefinition`	**Field Type** string **Description** Required. The API name of the disclosure definition associated with this version.
`documentTemplateGlobalKey`	**Field Type** string **Description** The document template global key associated with the DOCX template for the disclosure definition version.
`isActive`	**Field Type** boolean **Description** Indicates whether the disclosure definition version is an active version (`true`) or not (`false`). The default value is `false`.

Field Name	Description
`isCurrentVersion`	**Field Type** boolean **Description** Indicates whether this is the current version of the disclosure definition specified in the `disclosureDefinition` field (`true`) or not (`false`). The default value is `false`.
`isProtected`	**Field Type** boolean **Description** An auto-generated value that doesn't impact the behavior of the metadata type. The default is `false`.
`masterLabel`	**Field Type** string **Description** Required. A user-friendly name for DisclosureDefinitionVersion, which is defined when the DisclosureDefinitionVersion is created.
`omniScriptCnfgApiName`	**Description** The API name of the Omniscript configuration that's associated with the disclosure definition version. This field is required only when `authoringMode`isn't `Microsoft 365 Word`.
`omniScriptConfiguration`	**Field Type** string **Description** The ID of the Omniscript configuration record. Note: The value of this field is automatically populated using the API name of the OmniScript configuration specified in the `omniScriptCnfgApiName` field.
`versionNumber`	**Field Type** string **Description** Required. The version of the disclosure definition published by the author.

Declarative Metadata Sample Definition

The following is an example of a DisclosureDefinitionVersion component.

```
<?xml version="1.0" encoding="UTF-8"?>
<DisclosureDefinitionVersion xmlns="http://soap.sforce.com/2006/04/metadata">
    <description>This is GRI Disclosure Definition Version</description>
    <versionNumber>disclosure definition version number</versionNumber>
    <isActive>false</isActive>
    <disclosureDefinition>df10</disclosureDefinition>
    <omniScriptConfiguration>omni script configuration</omniScriptConfiguration>
    <omniScriptCnfgApiName>omni script config api name</omniScriptCnfgApiName>
    <isCurrentVersion>true</isCurrentVersion>
    <disclosureDefCurrVer>df10.Id</disclosureDefCurrVer>
    <documentTemplateGlobalKey>document template global key</documentTemplateGlobalKey>
    <authoringMode>OmniScriptForm</authoringMode>
    <masterLabel>GRI</masterLabel>
    <isProtected>false</isProtected>
</DisclosureDefinitionVersion>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package
 xmlns="http://soap.sforce.com/2006/04/metadata">
 <fullName>Pkg</fullName>
 <types>
  <members>GRI</members>
  <name>DisclosureDefinitionVersion</name>
 </types>
 <types>
  <members>df10</members>
  <name>DisclosureDefinition</name>
 </types>
 <types>
  <members>dt10</members>
  <name>DisclosureType</name>
 </types>
 <version>60.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

DisclosureType

Represents the types of disclosures that are done by an individual or an organization and the associated metadata.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

DisclosureType components have the suffix `.disclosureType` and are stored in the `disclosureTypes` folder.

Version

DisclosureType components are available in API version 57.0 and later.

Special Access Rules

The DisclosureAndComplianceHubAddOn license is required to access this object along with user access for the Disclosure Compliance Hub permission set license.

Fields

Field Name	Description
`description`	**Field Type** string **Description** The description about the disclosure type.
`disclosureBodyLogo`	**Field Type** string **Description** The logo ID of the standard body to which an individual or a company is making a disclosure.
`disclosureBodyUrl`	**Field Type** string **Description** The URL of the disclosure standard body.
`disclosureCategory`	**Field Type** string **Description** Required. The name of the clause category that's used for disclosure.

Field Name	Description
isProtected	**Field Type** boolean **Description** An auto-generated value that doesn't impact the behavior of the metadata type. The default is `false`.
masterLabel	**Field Type** string **Description** Required. A user-friendly name for DisclosureType, which is defined when the DisclosureType is created.

Declarative Metadata Sample Definition

The following is an example of a DisclosureType component.

```
<?xml version="1.0" encoding="UTF-8"?>
<DisclosureType
 xmlns="http://soap.sforce.com/2006/04/metadata">
 <description>This is ESG Disclosure Type</description>
 <disclosureBodyLogo>asdf</disclosureBodyLogo>
 <disclosureCategory>EnvSocGvnc</disclosureCategory>
 <disclosureBodyUrl>disclosure body url</disclosureBodyUrl>
 <isProtected>false</isProtected>
 <masterLabel>ESG</masterLabel>
</DisclosureType>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package
 xmlns="http://soap.sforce.com/2006/04/metadata">
 <fullName>Pkg</fullName>
 <types>
  <members>ESG</members>
  <name>DisclosureType</name>
 </types>
 <types>
  <name>StaticResource</name>
  <members>asdf</members>
 </types>
 <version>57.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character * (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

DiscoveryAIModel

Represents the metadata associated with a model used in Einstein Discovery.

A model is a sophisticated, custom algorithm that Einstein Discovery generates based on a comprehensive, statistical understanding of past outcomes. Einstein Discovery uses models to predict future outcomes. A model accepts the values of one or more predictor variables as input and produces a predicted outcome as output, along with (optionally) top factors and improvements. In Package Manager, this type is listed as "Discovery Model".

You can also build models using a third-party modeling tool, and then import them into Salesforce using Model Manager in Analytics Studio.

Note: Write operations for DiscoveryAIModel objects are generally not supported.

Declarative Metadata File Suffix and Directory Location

A DiscoveryAIModel is stored in the `discovery` folder. DiscoveryAIModels have two files:

- file with `.model` suffix contains the model's actual data
- file named *ModelName*`.model-meta.xml` suffix contains the model's metadata

Here is a sample `package.xml` file:

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>Maximize_Sales</members>
        <name>DiscoveryAIModel</name>
    </types>
    <version>53.0</version>
</Package>
```

Version

DiscoveryAIModels are available in API version 51.0 and later.

Fields

Field Name	Field Type	Description
`algorithmType`	DiscoveryAlgorithmType	Algorithm that Einstein Discovery used to create the model associated with this story.
`classificationThreshold`	double	Threshold value. Applies only to binary classification models. For regression models, this is null.

Field Name	Field Type	Description
`description`	string	Model description.
`label`	string	Model label. If you package a model, this label appears in Package Manager.
`modelFields`	DiscoveryModelField[]	One or more model fields (variables).
`modelRuntimeType`	DiscoveryModelRuntimeType	Model run-time type.
`predictedField`	string	Name of the field that is predicted.
`predictionType`	DiscoveryPredictionType	Type of prediction. One of the following strings: • `Regression` • `Classification` • `Unknown`
`sourceType`	DiscoveryModelSourceType	Source type.
`status`	DiscoveryAIModelStatus	Model status (enabled or disabled).
`trainingMetrics`	string	JSON object that represents metrics about the model when it was trained.
`transformations`	DiscoveryModelTransform	One or more model transformations.

DiscoveryAlgorithmType

Represents the algorithm that Einstein Discovery used to create the model.

Field Name	Field Type	Description
`Best`	string	Tournament Model. Genetic algorithm used to generate high-quality solutions to optimization and search problems, like optimizing decision trees for better performance.
`Glm`	string	Generalized Linear Model. Regression-based algorithm.
`Gbm`	string	Gradient Boost Machine. Decision tree-based ensemble machine learning algorithm.
`Xgboost`	string	XGBoost. Decision tree-based ensemble machine learning algorithm.
`Drf`	string	Random Forest. Supervised learning algorithm that uses multiple decision trees, randomization, and other optimization techniques.

DiscoveryModelField

Represents a field (variable) in the model.

Field Name	Field Type	Description
`isDisparateImpact`	boolean	Indicates whether the field is disparate impact (`true`) or not (`false`).
`isSensitive`	boolean	Indicates whether the field is sensitive (`true`) or not (`false`).
`label`	string	Field label displayed in the UI.
`name`	string	Field name.
`type`	DiscoveryModelFieldType	Field type. Enumerated.
`values`	string[]	A list of field values.

DiscoveryModelTransform

Represents a transformation in the model.

Field Name	Field Type	Description
`config`	string	The configuration for the transformation.
`sourceFieldNames`	string[]	A list of the source field names.
`targetFieldNames`	string[]	A list of the target field names.
`type`	DiscoveryAIModelTransformationType	Type of transformation.

DiscoveryAIModelTransformationType

Represents the type of transformation to apply before making a prediction.

Field Name	Field Type	Description
`TypographicClustering`	string	Typographic clustering transformation.
`SentimentAnalysis`	string	Sentiment analysis transformation.
`FreeTextClustering`	string	Free text clustering transformation.
`NumericalImputation`	string	Numerical imputation transformation.
`CatagoricalImputation`	string	Catagorical imputation transformation.
`TimeSeriesForecast`	string	Time series forecast transformation.
`ExtractMonthOfYear`	string	Extract month of year transformation.
`ExtractDayOfWeek`	string	Extract day of week transformation.
`ZipCodeAnalysis`	string	Zip code analysis transformation.

DiscoveryModelFieldType

Represents the data type of a model field.

Field Name	Field Type	Description
`Text`	string	Text data type.
`Number`	string	Number data type.
`Date`	string	Date data type.

DiscoveryModelRuntimeType

Represents the model run-type.

Field Name	Field Type	Description
`Discovery`	string	The model run-type is Einstein Discovery.
`H2O`	string	The model run-type is H20.
`T`	string	The model run-type is Tensorflow v2.4.4.
`Tf27`	string	The model run-type is Tensorflow v2.7.0.
`SC102`	string	The model run-type is Scikit Learn v1.0.2.

DiscoveryModelSourceType

Represents the source tool used to build the model: Discovery or an external tool (the model was uploaded into Salesforce).

Field Name	Field Type	Description
`Discovery`	string	Einstein Discovery built the model.
`UserUpload`	string	An external tool built the model. The model was then uploaded into Salesforce. Note: This source type is not supported in the Metadata API.

DiscoveryAIModelStatus

Represents the status of the model (Enabled or Disabled).

Field Name	Field Type	Description
`Disabled`	string	The model is disabled (inactive).
`Uploading`	string	The model is uploading.

Field Name	Field Type	Description
UploadFailed	string	The model failed to upload.
UploadCompleted	string	The model upload is complete.
Validating	string	The model is validating.
ValidationFailed	string	The model validation failed.
ValidationCompleted	string	The model validation is complete.
Enabled	string	The model is enabled (active).

Declarative Metadata Sample Definitions

Here is a sample DiscoveryAIModel:

```
<?xml version="1.0" encoding="UTF-8"?>
<DiscoveryAIModel xmlns="http://soap.sforce.com/2006/04/metadata"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <content xsi:nil="true"/>
   <algorithmType>Glm</algorithmType>
   <classificationThreshold>0.7383</classificationThreshold>
   <label>Maximize Tenure</label>
   <modelFields>
      <label>Field</label>
      <name>Field</name>
      <type>Text</type>
   </modelFields>
   <modelFields>
      <label>PTO</label>
      <name>PTO</name>
      <type>Number</type>
   </modelFields>
   <modelFields>
      <label>Level</label>
      <name>Level</name>
      <type>Text</type>
   </modelFields>
   <modelFields>
      <label>Salary</label>
      <name>Salary</name>
      <type>Number</type>
   </modelFields>
   <modelFields>
      <label>Tenure</label>
      <name>Tenure</name>
      <type>Number</type>
   </modelFields>
   <modelRuntimeType>Discovery</modelRuntimeType>
   <predictedField>Tenure</predictedField>
   <predictionType>Classification</predictionType>
   <sourceType>Discovery</sourceType>
```

```
    <status>Enabled</status>
</DiscoveryAIModel>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

DiscoveryGoal

Represents the metadata associated with an Einstein Discovery prediction definition.

A prediction definition is a container object in Einstein Discovery that is associated with one or more deployed models. If a prediction definition contains multiple models, then each model produces predictions for a different segment of the data. A prediction definition can contain up to ten active models. In Package Manager, this type is listed as "Discovery Prediction".

Declarative Metadata File Suffix and Directory Location

A DiscoveryGoal is stored in the `discovery` folder. DiscoveryGoals have a `.goal` file suffix. Here is a sample `package.xml` file:

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>employees_Tenure</members>
        <name>DiscoveryGoal</name>
    </types>
    <version>53.0</version>
</Package>
```

Version

DiscoveryGoals are available in API version 51.0 and later.

Fields

Field Name	Field Type	Description
`active`	boolean	Indicates whether the prediction definition is active (True) or not (False).
`deployedModels`	DiscoveryDeployedModel[]	One or more deployed models associated with this prediction definition.
`label`	string	Name of the prediction definition.
`modelCards`	DiscoveryModelCard[]	Model card for this prediction definition.
`outcome`	DiscoveryGoalOutcome	Outcome variable of this prediction definition.

Field Name	Field Type	Description
predictionType	DiscoveryPredictionType	Type of prediction: Regression, Classification, or Unknown.
pushbackField	string	Automated writeback field for predictions. A custom field on the Salesforce object specified in subscribedEntity. Note: Removing a pushback field from the goal metadata causes the field to be deleted from the Salesforce object as well.
pushbackType	DiscoveryPushbackType	Type of writeback field for predictions.
subscribedEntity	string	Salesforce object associated with this model.
terminalStateFilters	DiscoveryFilter[]	If specified, one or more filter expressions that define the conditions under which an observation has attained its terminal state (the actual outcome has been reached). For performance monitoring, Einstein Discovery determines model accuracy by comparing a model's predicted outcomes with actual (observed) outcomes.

DiscoveryDeployedModel

Represents a model deployed in Salesforce.

Field Name	Field Type	Description
active	boolean	Indicates whether the deployed model is active (True) or inactive (False).
aiModel	string	Full name of the DiscoveryAIModel being deployed.
classificationThreshold	double	Threshold value. Applies only to binary classification models. For regression models, this is null.
fieldMappings	DiscoveryFieldMap[]	One or more mappings between model variables and either fields (in Salesforce objects) or columns (in CRM Analytics datasets).
filters	DiscoveryFilter[]	If specified, one or more segmentation filters for the deployed model. When making a prediction, the first model that has filters matching a specific input row will be used to make the prediction. No filters indicates that the model matches all input rows.
label	string	Label for the deployed model. Appears in Model Manager.
name	string	Name of the deployed model.
prescribableFields	DiscoveryPrescribableField[]	Actionable fields associated with improvements.

DiscoveryFieldMap

Represents a mapping between model variables and field values.

Field Name	Field Type	Description
`mappedField`	string	Field in a Salesforce object or column in a CRM Analytics dataset.
`modelField`	string	Model variable.
`sobjectFieldJoinKey`	string	Join key for a Salesforce object. Null if `sourceType` is `AnalyticsDatasetField`.
`source`	string	If the mapping is to a CRM Analytics dataset, this is the name of the dataset. Otherwise, null.
`sourceFieldJoinKey`	string	If the mapping is to a CRM Analytics dataset, this is the lookup column on that dataset used to perform the join. Otherwise, null.
`sourceType`	DiscoveryFieldMapSourceType	Data source type for field mapping.

DiscoveryFieldMapSourceType

Represents the data source type for field mapping: `SalesforceField` or `AnalyticsDatasetField`.

Field Name	Field Type	Description
`SalesforceField`	string	Field in a Salesforce object.
`AnalyticsDatasetField`	string	Column in a CRM Analytics dataset.

DiscoveryFilter

Represents a field filter.

Field Name	Field Type	Description
`field`	string	Name of the field to filter.
`operator`	DiscoveryFilterOperator	Operator used to calculate the filter.
`type`	DiscoveryFilterFieldType	Type of filter value.
`values`	DiscoveryFilterValue[]	One or more values selected for the filter.

DiscoveryFilterOperator

Represents a filter operator.

Field Name	Field Type	Description
`Equal`	string	Equal to operator (=).

Field Name	Field Type	Description
NotEqual	string	Not equal to operator (<>).
GreaterThan	string	Greater than operator (>).
GreaterThanOrEqual	string	Greater than or equal to operator (>=).
LessThan	string	Less than operator (<).
LessThanOrEqual	string	Less than or equal to operator (<=).
Between	string	Between operator.
NotBetween	string	Not between operator.
InSet	string	In set operator.
NotIn	string	Not in operator.
Contains	string	Contains operator.
StartsWith	string	Starts with operator.
EndsWith	string	Ends with operator.
IsNull	string	Is null operator.
IsNotNull	string	Is not null operator.

DiscoveryFilterFieldType

Represents the data type of the filter field.

Field Name	Field Type	Description
Text	string	Text field type.
Number	string	Number field type.
Date	string	Date field type.
DateTime	string	Datetime field type.
Boolean	string	Boolean field type.

DiscoveryFilterValue

Represents a filter value.

Field Name	Field Type	Description
type	DiscoveryFilterValueType	Type of filter value.
value	DiscoveryFilterValue	Value.

DiscoveryFilterValueType

Represents the type of filter value.

Field Name	Field Type	Description
`Constant`	string	Filter value is a constant.
`PlaceHolder`	string	Filter value is a placeholder.

DiscoveryPrescribableField

Represents custom improvement text.

Field Name	Field Type	Description
`customDefinitions`	DiscoveryCustomPrescribableFieldDefinition[]	One or more strings for custom improvement text. Uses the default improvement text if none are specified.
`name`	string	Name of the model field that is actionable.

DiscoveryCustomPrescribableFieldDefinition

Represents a field definition in custom improvement text.

Field Name	Field Type	Description
`filters`	DiscoveryFilter[]	Represents one or more filters associated with custom improvement text.
`template`	string	If specified, represents the user-provided template from which the custom text is computed. If not specified, then the default text is used.

DiscoveryModelCard

Represents a model card associated with an Einstein Discovery prediction definition.

Field Name	Field Type	Description
`contactEmail`	string	Contact email for this model card.
`contactName`	string	Contact name for this model card.
`label`	string	Title for this model card.
`sections`	string	Sections in the model card.

DiscoveryGoalOutcome

Represents the outcome variable of the model.

Field Name	Field Type	Description
`field`	string	Name of the outcome variable.
`fieldLabel`	string	Label for the outcome variable.
`goal`	DiscoveryOutcomeGoal	Goal for the outcome variable.
`mappedField`	string	Mapped field.

DiscoveryOutcomeGoal

Represents the goal for an outcome.

Field Name	Field Type	Description
`Minimize`	string	Maximize the outcome.
`Maximize`	string	Minimize the outcome.
`None`	string	Reserved for future use.

DiscoveryPredictionType

Represents the prediction type for a model.

Field Name	Field Type	Description
`Unknown`	string	Unknown prediction type.
`Regression`	string	Regression prediction (numeric use case).
`Classification`	string	Binary classification prediction.
`MulticlassClassification`	string	Multiclass classification prediction.

DiscoveryPushbackType

Represents the type of writeback field. Must be set to `AiRecordInsight`.

Field Name	Field Type	Description
`AiRecordInsight`	string	Automatic writeback type. Required.
`Direct`	string	Currently not supported. Reserved for future use.

Declarative Metadata Sample Definitions

Here is a sample DiscoveryGoal:

```
<?xml version="1.0" encoding="UTF-8"?>
<DiscoveryGoal xmlns="http://soap.sforce.com/2006/04/metadata"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <active>true</active>
   <deployedModels>
      <active>true</active>
      <aiModel>Maximize_Tenure</aiModel>
      <fieldMappings>
         <mappedField>Opportunity.Amount</mappedField>
         <modelField>PTO</modelField>
         <sourceType>SalesforceField</sourceType>
      </fieldMappings>
      <fieldMappings>
         <mappedField>Opportunity.ExpectedRevenue</mappedField>
         <modelField>Salary</modelField>
         <sourceType>SalesforceField</sourceType>
      </fieldMappings>
      <fieldMappings>
         <mappedField>Level</mappedField>
         <modelField>Level</modelField>
         <sobjectFieldJoinKey>Opportunity.Name</sobjectFieldJoinKey>
         <source>employees</source>
         <sourceFieldJoinKey>Name</sourceFieldJoinKey>
         <sourceType>AnalyticsDatasetField</sourceType>
      </fieldMappings>
      <fieldMappings>
         <mappedField>Opportunity.StageName</mappedField>
         <modelField>Field</modelField>
         <sourceType>SalesforceField</sourceType>
      </fieldMappings>
      <filters>
         <field>Opportunity.StageName</field>
         <operator>Equal</operator>
         <values>
            <type>Constant</type>
            <value>Qualification</value>
         </values>
      </filters>
      <label>employees</label>
      <name>employees</name>
      <prescribableFields>
         <customDefinitions>
           <filters>
              <field>Salary</field>
              <operator>LessThan</operator>
              <type>Number</type>
              <values>
                 <type>PlaceHolder</type>
                 <value>[value_low]</value>
              </values>
```

```
          </filters>
          <template>Increase [field_name] by [diff]</template>
       </customDefinitions>
       <customDefinitions>
          <filters>
             <field>Salary</field>
             <operator>GreaterThan</operator>
             <type>Number</type>
             <values>
                 <type>PlaceHolder</type>
                 <value>[value_low]</value>
              </values>
           </filters>
           <template xsi:nil="true"/>
        </customDefinitions>
        <name>Salary</name>
    </prescribableFields>
    <prescribableFields>
        <customDefinitions>
           <filters>
              <field>Level</field>
              <operator>LessThan</operator>
              <type>Number</type>
              <values>
                 <type>PlaceHolder</type>
                 <value>[value_low]</value>
              </values>
         </filters>
         <template xsi:nil="true"/>
        </customDefinitions>
        <customDefinitions>
           <filters>
              <field>Level</field>
              <operator>GreaterThan</operator>
              <type>Number</type>
              <values>
                 <type>PlaceHolder</type>
                 <value>[value_low]</value>
               </values>
           </filters>
           <template xsi:nil="true"/>
        </customDefinitions>
        <name>Level</name>
    </prescribableFields>
    <prescribableFields>
       <name>Field</name>
    </prescribableFields>
</deployedModels>
<label>employees_Tenure</label>
<outcome>
    <field>Tenure</field>
    <fieldLabel>Tenure</fieldLabel>
    <goal>Maximize</goal>
    <mappedField>Opportunity.Amount</mappedField>
```

```
    </outcome>
    <predictionType>Regression</predictionType>
    <pushbackField>My_Pushback_Field__c</pushbackField>
    <subscribedEntity>Opportunity</subscribedEntity>
    <terminalStateFilters>
        <field>Opportunity.Amount</field>
        <operator>GreaterThan</operator>
        <values>
            <type>Constant</type>
            <value>5</value>
        </values>
    </terminalStateFilters>
    <terminalStateFilters>
        <field>Opportunity.Amount</field>
        <operator>LessThan</operator>
        <values>
            <type>Constant</type>
            <value>10</value>
        </values>
    </terminalStateFilters>
</DiscoveryGoal>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

DiscoveryStory

Represents the metadata associated with a story used in Einstein Discovery.

A story defines the data and analytical settings that Einstein Discovery uses to generate insights and build predictive models. Story settings include the outcome variable, whether to maximize or minimize the outcome variable, the data to analyze in a CRM Analytics dataset, and other preferences. Story settings tell Einstein Discovery how to conduct the analysis and communicate its results. In Package Manager, this type is listed as "Discovery Story".

Note: Write operations for DiscoveryStory objects are generally not supported.

Declarative Metadata File Suffix and Directory Location

A DiscoveryStory is stored in the `discovery` folder. DiscoveryStory have two files:

- file with `.story` suffix contains the story's actual data
- file named *`ModelName`*`.story-meta.xml` suffix contains the story's metadata

Here is a sample `package.xml` file:

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>Subscriber_Changes</members>
        <name>DiscoveryStory</name>
```

```
    </types>
    <version>55.0</version>
</Package>
```

Version

DiscoveryStorys are available in API version 54.0 and later.

Fields

Field Name	Field Type	Description
`application`	string	Required. The CRM Analytics app the story is associated with.
`autopilot`	DiscoveryStoryAutopilotStatus	Optional. The autopilot status for the story. One of the following strings: • `Enabled` • `Disabled`
`classificationThreshold`	double	Optional. The threshold for classification predictions for the story.
`label`	string	Required. The story label. If you package a story, this label appears in Package Manager.
`outcome`	DiscoveryStoryOutcome	Required. The selected outcome of the story.
`sourceContainer`	string	Required. The source ID for the story.
`sourceType`	DiscoveryStorySourceType	Required. The source type of the story. One of the following strings: • `AnalyticsDataset` • `LiveDataset` • `Report`
`validationContainder`	string	Optional. The validation ID for the story.

DiscoveryStoryOutcome

Represents the selected outcome of the generated story.

Field Name	Field Type	Description
`failureValue`	string	Optional. The value if the story failed.
`field`	string	Required. The field configuration for the story.
`goal`	DiscoveryStoryOutcomeGoal	Required. The story outcome goal. One of the following strings: • `Maximize` • `Minimize`

Field Name	Field Type	Description
		• None
label	string	Required. The story outcome label.
successValue	string	Optional. The value if the story succeeded.
type	DiscoveryStoryOutcomeType	Required. The story outcome type. One of the following strings: • Categorical • Count • Number • Text

Declarative Metadata Sample Definitions

Here is a sample DiscoveryStory:

```
<?xml version="1.0" encoding="UTF-8"?>
<DiscoverStory xmlns="http://soap.sforce.com/2006/04/metadata"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <application>MyStoryApp</application>
    <autopilot>Enabled</autopilot>
    <classificationThreshold>0.7383</classificationThreshold>
    <label>SubscriberChanges</label>
    <outcome>
        <field>Subscriber</field>
        <goal>Minimize</goal>
        <label>SubscriberChangeOutcome</label>
        <successValue>Success</successValue>
        <type>Numerical</type>
    </outcome>
    <sourceContainer>01X00000000xxxx1AB</sourceContainer>
    <sourceType>AnalyticsDataset</sourceType>
</DiscoveryStory>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character * (asterisk) in the package.xml manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

Document

Represents a Document. All documents must be in a document folder, such as sampleFolder/TestDocument.

This type extends the MetadataWithContent metadata type and inherits its content and fullName fields.

Retrieving Documents

You can't use the wildcard (*) symbol with documents in `package.xml`. To retrieve the list of documents for populating `package.xml` with explicit names, call `listMetadata()` and pass in `DocumentFolder` as the type. Note that DocumentFolder is not returned as a type in `describeMetadata()`. Document is returned from `describeMetadata()` with an associated attribute of `inFolder` set to true. If that attribute is set to true, you can construct the type by using the component name with the word Folder, such as DocumentFolder.

The following example shows folders in `package.xml`:

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>MyDBFolder/MyDBName</members>
        <name>Dashboard</name>
    </types>
    <types>
        <members>MyDocumentFolder/MyDocumentName</members>
        <name>Document</name>
    </types>
    <types>
        <members>unfiled$public/MarketingProductInquiryResponse</members>
        <members>unfiled$public/SalesNewCustomerEmail</members>
        <name>EmailTemplate</name>
    </types>
    <types>
        <members>MyReportFolder/MyReportName</members>
        <name>Report</name>
    </types>
    <version>64.0</version>
</Package>
```

For each document an accompanying metadata file named *`DocumentFilename`*`-meta.xml` is created in the document folder. For example, for a document `TestDocument.png` in the sampleFolder folder, there's a `TestDocument.png-meta.xml` in the `documents/sampleFolder` of the package.

Version

Documents are available in API version 10.0 and later.

In API version 17.0 and later, you can delete a folder containing documents moved to the Recycle Bin. When you delete the folder, any related documents in the Recycle Bin are permanently deleted.

In API version 18.0 and later, documents do not need an extension.

Fields

This metadata type contains the following fields:

Field Name	Field Type	Description
`content`	base64	Content of the document. Base 64-encoded binary data. Prior to making an API call, client applications must encode the binary attachment data as base64. Upon receiving a response, client applications must decode

Field Name	Field Type	Description
		the base64 data to binary. This conversion is usually handled for you by a SOAP client. This field is inherited from the MetadataWithContent component.
`description`	string	A description of the document. Enter a description to distinguish this document from others.
`fullName`	string	The name of the document, including the folder name. In version 17.0 and earlier, the `fullName` included the document extension. In version 18.0 and later, the `fullName` does not include the file extension. The `fullName` can contain only underscores and alphanumeric characters. It must be unique, begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. If this field contained characters before version 14.0 that are no longer allowed, the characters were stripped out of this field, and the previous value of the field was saved in the `name` field. This field is inherited from the Metadata component.
`internalUseOnly`	boolean	Required. Indicates whether the document is confidential (`true`) or not (`false`). This field and `public` are mutually exclusive; you cannot set both to `true`.
`keywords`	string	Contains one or more words that describe the document. A check for matches to words in this field is performed when doing a search.
`name`	string	The list of characters allowed in the `fullName` field has been reduced for versions 14.0 and later. This field contains the value contained in the `fullName` field before version 14.0. This field is only populated if the value of the `fullName` field contained characters that are no longer accepted in that field.
`public`	boolean	Required. Indicates whether the document is an image available for HTML email templates and does not require a Salesforce username and password to view in an email (`true`) or not (`false`). If the images will be used as a custom app logo or custom tab icon, both of which require a Salesforce username and password to view, set this field to `false`. This field and `internalUseOnly` are mutually exclusive; you cannot set both to `true`.

Declarative Metadata Sample Definition

The following is the definition of a document:

```
<?xml version="1.0" encoding="UTF-8"?>
<Document xmlns="http://soap.sforce.com/2006/04/metadata">
    <internalUseOnly>false</internalUseOnly>
    <name>Q2 Campaign Analysis</name>
    <public>false</public>
    <description>Analyze Q2 campaign effectiveness</description>
</Document>
```

Wildcard Support in the Manifest File

This metadata type doesn't support the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SEE ALSO:

Folder

DocumentCategory

Represents a document category.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

DocumentCategory components have the suffix `.documentCategory` and are stored in the `documentCategory` folder.

Version

DocumentCategory components are available in API version 59.0 and later.

Special Access Rules

Fields

Field Name	Description
`description`	**Field Type** string **Description** A description of the DocumentCategory.
`isProtected`	**Field Type** boolean **Description** An auto-generated value that doesn't impact the behavior of the metadata type. The default value is `false`.
`masterLabel`	**Field Type** string

Field Name	Description
	Description Required. The master label of the DocumentCategory. This internal label doesn't get translated.

Declarative Metadata Sample Definition

The following is an example of a DocumentCategory component.

```
<?xml version="1.0" encoding="UTF-8"?>
<DocumentCategory xmlns="http://soap.sforce.com/2006/04/metadata">
    <masterLabel>Address_Proof</masterLabel>
</DocumentCategory>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>DocumentCategory</name>
    </types>
    <version>59.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

DocumentCategoryDocumentType

Represents the junction between a DocumentCategory and a DocumentType. Puts a DocumentType in a DocumentCategory.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

DocumentCategoryDocumentType components have the suffix `.documentCategoryDocumentType` and are stored in the `documentCategoryDocumentTypes` folder.

Version

DocumentCategoryDocumentType components are available in API version 59.0 and later.

Special Access Rules

Fields

Field Name	Description
`documentCategory`	**Field Type** string **Description** Required. The master label of the related DocumentCategory.
`documentType`	**Field Type** string **Description** Required. The master label of the related DocumentType.
`isProtected`	**Field Type** boolean **Description** An auto-generated value that doesn't impact the behavior of the metadata type. The default value is `false`.
`masterLabel`	**Field Type** string **Description** Required. The master label of the DocumentCategoryDocumentType. This internal label doesn't get translated.

Declarative Metadata Sample Definition

The following is an example of a DocumentCategoryDocumentType component.

```
<?xml version="1.0" encoding="UTF-8"?>
<DocumentCategoryDocumentType xmlns="http://soap.sforce.com/2006/04/metadata">
    <documentCategory>Address_Proof</documentCategory>
    <documentType>Utility_Bill</documentType>
    <masterLabel>junction1</masterLabel>
</DocumentCategoryDocumentType>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>DocumentCategory</name>
    </types>
    <types>
        <members>*</members>
        <name>DocumentCategoryDocumentType</name>
    </types>
    <types>
        <members>*</members>
        <name>DocumentType</name>
    </types>
    <version>59.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

DocumentChecklistSettings

Represents an org's DocumentChecklistItem settings. This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for more details.

File Suffix and Directory Location

DocumentChecklistSettings components are stored in the `DocumentChecklist.settings` file in the `settings` folder. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

DocumentChecklistSettings components are available in API versions 55.0 and later.

Fields

Field Name	Field Type	Description
`dciCustomSharing`	boolean	Indicates whether the custom sharing rule for document checklist items is enabled for your org (`true`) or not (`false`). The default value is `false`.
`deleteDCIWithFiles`	boolean	Indicates whether deletion of document checklist items is enabled for your org (`true`) or not (`false`). The default value is `false`.

Declarative Metadata Sample Definition

The following is an example of a DocumentChecklistSettings.settings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<DocumentChecklistSettings
 xmlns="http://soap.sforce.com/2006/04/metadata">
 <dciCustomSharing>true</dciCustomSharing>
 <deleteDCIWithFiles>true</deleteDCIWithFiles>
</DocumentChecklistSettings>
```

Example Package Manifest

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package
 xmlns="http://soap.sforce.com/2006/04/metadata">
 <types>
  <members>DocumentChecklist</members>
  <name>Settings</name>
 </types>
 <version>55.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

DocumentType

Represents a document type.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

DocumentType components have the suffix `.documentType` and are stored in the `documentTypes` folder.

Version

DocumentType components are available in API version 59.0 and later.

Special Access Rules

Fields

Field Name	Description
`description`	**Field Type** string **Description** A description of the DocumentType.
`isActive`	**Field Type** boolean **Description** Required. Specifies whether the DocumentType is active.
`masterLabel`	**Field Type** string **Description** Required. The master label of the DocumentType. This internal label doesn't get translated.

Declarative Metadata Sample Definition

The following is an example of a DocumentType component.

```
<?xml version="1.0" encoding="UTF-8"?>
<DocumentType xmlns="http://soap.sforce.com/2006/04/metadata">
    <description>Utility_Bill</description>
    <isActive>true</isActive>
    <masterLabel>Utility_Bill</masterLabel>
</DocumentType>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>DocumentType</name>
    </types>
    <version>59.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

DuplicateRule

Represents a rule that specifies how duplicate records in an object are detected. This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

DuplicateRule components have the `.duplicateRule` suffix and are stored in the `duplicateRules/` directory. The name of the component file is based on the name of the object associated with the rule. For example, the component file name `duplicateRules/Account.Standard_Account_Duplicate_Rule.duplicateRule` describes a duplicate rule component associated with the Account object.

Version

DuplicateRule components are available in API version 64.0 and later.

Fields

Field Name	Field Type	Description
`actionOnInsert`	DupeActionType (enumeration of type string)	Required. Determines what the duplicate rule does when users or the DuplicateRule API try to insert a record that is a duplicate. Valid values are: **`Allow`** For users, if `operationsOnInsert` is set to `alert`, the UI displays the value of `alertText` in a dialog. The dialog prompts users to continue or cancel. If the user chooses to continue, the insertion proceeds. If the user chooses to cancel, the record isn't inserted. The DuplicateRule API returns an error code and a message. To complete the insertion, the code must set the `allowSave` field in DuplicateRuleHeader to `true` and reissue the request. If `operationsOnInsert`isn't set to `alert`, the UI inserts the record without issuing an alert. The API inserts the record and doesn't return an error code.

Field Name	Field Type	Description
		`Block` For users, the UI displays an error message and prevents them from inserting the new record. The DuplicateRule API returns an error and doesn't insert the record.
`actionOnUpdate`	DupeActionType (enumeration of type string)	Required. Determines what the duplicate rule does when users or the DuplicateRule API try to update a record, and the result is a duplicate. Valid values are: **`Allow`** For users, if `operationsOnUpdate` is set to `alert`, the UI displays the value of `alertText` in a dialog. The dialog prompts users to continue or cancel. If the user chooses to continue, the update proceeds. If the user chooses to cancel, the record isn't updated. The DuplicateRule API returns a message. To complete the update, the code must set the `allowSave` field in DuplicateRuleHeader to `true` and reissue the request. If `operationsOnUpdate`isn't set to `alert`, the UI updates the record without issuing an alert. The API updates the record and doesn't return an error code. **`Block`** For users, the UI displays an error message and prevents them from continuing. The DuplicateRule API returns an error.
`alertText`	string	Text that's sent when the duplicate rule is triggered. The text is only sent if `isActive` is `true`. In the UI, the text displays as a message. The DuplicateRule API returns the message in its response. You can set a value for `alertText` only when you have `actionOnInsert` or `actionOnUpdate` (or both) set to `Allow`. Otherwise, you receive a validation error when you add or update this component.
`description`	string	Required. Text that describes the duplicate rule. The value is customer-supplied, but isn't visible in the UI.
`duplicateRuleFilter`	DuplicateRuleFilter	Required. Criteria that define how to find records to consider when looking for duplicates. For example, use `duplicateRuleFilter` to exclude records from the match when looking for duplicates.
`duplicateRuleMatchRules`	DuplicateRuleMatchRule[]	Required. One or more MatchingRule components for the DuplicateRule. A `MatchingRule` controls what constitutes a match between records.
`isActive`	boolean	Required. If `true`, the DuplicateRule detects duplicate records. Otherwise, the rule has no effect.
`masterLabel`	string	Required. Label for this DuplicateRule. This value is the internal label for the rule.

Field Name	Field Type	Description
`operationsOnInsert`	string[]	Required. Controls the action to take when `actionOnInsert` is set to `Allow` and the duplicate rule is triggered. Either one or both of these values can be set in the array: **`alert`** If set, the action specified in `actionOnInsert` occurs; otherwise, the insert proceeds. **`report`** If set, the insert operation is added to the report of duplicates.
`operationsOnUpdate`	string[]	Required. Controls the action to take when `actionOnUpdate` is set to `Allow` and the duplicate rule is triggered. Either one or both of these values can be set in the array: **`alert`** If set, the action specified in `actionOnUpdate` occurs; otherwise, the update proceeds. **`report`** If set, the update operation is added to the report of duplicates.
`securityOption`	DupeSecurityOptionType (enumeration of type string)	Required. Determines how record sharing rules affect duplicate management. Valid values are: **`EnforceSharingRules`** Sharing rules affect duplicate management. If a duplicate rule is triggered because an insert or update duplicates an existing record, but the running user doesn't have sharing access to that record, the insert or update proceeds. The sharing rule doesn't prevent the user from creating or updating the record because the record is hidden from the user. No message is issued. **`BypassSharingRules`** Sharing rules don't affect duplicate management. If a duplicate rule is triggered because an insert or update duplicates an existing record, sharing rules are ignored, but other access restrictions apply.
`sortOrder`	int	Required. Determines the order in which duplicate rules are applied.

DuplicateRuleMatchRule

Describes the MatchingRule associated with the `DuplicateRule`. The `MatchingRule` identifies duplicate records.

Field Name	Field Type	Description
`matchRuleSObjectType`	string	Required. The name of the target object of the matching rule. For example, if you define a duplicate rule for Contact records, and you want to match with Lead records, the value of `matchRuleSObjectType` is Lead.

Field Name	Field Type	Description
`matchingRule`	string	Required. Value that corresponds to the value of `developerName` in the MatchingRule for this duplicate rule.
`objectMapping`	ObjectMapping	Required. Foreign key to an ObjectMapping that maps fields from the duplicate rule's object to fields in the target object specified by `matchRuleSObjectType`.

DuplicateRuleFilter

Specifies filter criteria for a DuplicateRule. Salesforce only applies the DuplicateRule if the record matches the criteria.

Field Name	Field Type	Description
`booleanFilter`	string	Required. A string of boolean operators that establishes the filter logic for the filter items specified in `duplicateRuleFilterItems`.
`duplicateRuleFilterItems`	DuplicateRuleFilterItem[]	Required. A list of DuplicateRuleFilterItem components.

DuplicateRuleFilterItem

This type extends the FilterItem type and inherits all its fields.

Field Name	Field Type	Description
`sortOrder`	int	Required. The order of this item in the duplicate rule filter.
`table`	string	Required. The object that has the field specified in the `field` field of DuplicateRuleFilterItem. See the documentation for FilterItem for the definition of `field`.

ObjectMapping

Represents a map of fields in the input object of the DuplicateRule to fields in the output object of DuplicateRule. The input object is the object associated with the DuplicateRule. The output object can be the same object or a different object with similar fields.

For example, you can have a DuplicateRule that looks for duplicates between the Contact object and the Lead object. In this case, the input object is Contact, and the output object is Lead.

Field Name	Field Type	Description
`inputObject`	string	Required. The input object for the duplicate rule. The DuplicateRule is associated with this object. For example, if you define a duplicate rule for Contact records, and you want to match with Lead records, the value of `inputObject` is Contact.
`mappingFields`	ObjectMappingField[]	Required. The mapping of source object fields to target object fields for the duplicate rule.

Field Name	Field Type	Description
outputObject	string	Required. The output object for the duplicate rule. This value is the same as the value of the matchRuleSObjectType field in DuplicateRuleMatchRule. Any duplicate rules that this object has are ignored when the DuplicateRule API uses the ObjectMapping.

ObjectMappingField

A field name in the input object of the DuplicateRule, and the corresponding field name in the output object.

Field Name	Field Type	Description
inputField	string	Required. Field in the object specified by the inputObject field in ObjectMapping. This field is mapped to the field in outputField, which is assumed to be a field in the object specified by the outputObject field in ObjectMapping.
outputField	string	Required. Field in the object specified by the outputObject field in ObjectMapping. The field is mapped to the field name in inputField, which is assumed to be a field in the object specified by the inputObject in ObjectMapping.

Declarative Metadata Sample Definition

The following is an example of a DuplicateRule component.

```
<?xml version="1.0" encoding="UTF-8"?>
<DuplicateRule xmlns="http://soap.sforce.com/2006/04/metadata"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <actionOnInsert>Allow</actionOnInsert>
    <actionOnUpdate>Allow</actionOnUpdate>
    <alertText>You are creating a duplicate record. Use an existing record
instead.</alertText>
    <description>Detects a contact that duplicates a Lead</description>
    <duplicateRuleFilter>
        <booleanFilter xsi:nil="true"/>
        <duplicateRuleFilterItems>
            <field>Username</field>
            <operation>equals</operation>
            <value>user@example.com</value>
            <sortOrder>1</sortOrder>
            <table>User</table>
        </duplicateRuleFilterItems>
    </duplicateRuleFilter>
    <duplicateRuleMatchRules>
        <matchRuleSObjectType>Lead</matchRuleSObjectType>
        <matchingRule>ContactToLeadDuplicate_matching_rule</matchingRule>
        <objectMapping>
            <inputObject>Contact</inputObject>
            <mappingFields>
```

```
                <inputField>FirstName</inputField>
                <outputField>FirstName</outputField>
            </mappingFields>
            <mappingFields>
                <inputField>LastName</inputField>
                <outputField>LastName</outputField>
            </mappingFields>
            <outputObject>Lead</outputObject>
        </objectMapping>
    </duplicateRuleMatchRules>
    <isActive>true</isActive>
    <masterLabel>ContactToLeadDuplicate</masterLabel>
    <operationsOnInsert>Alert</operationsOnInsert>
    <operationsOnInsert>Report</operationsOnInsert>
    <operationsOnUpdate>Alert</operationsOnUpdate>
    <operationsOnUpdate>Report</operationsOnUpdate>
    <securityOption>EnforceSharingRules</securityOption>
    <sortOrder>1</sortOrder>
</DuplicateRule>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>ContactToLeadDuplicate</members>
        <name>DuplicateRule</name>
    </types>
    <version>38.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

EclairGeoData

Represents an Analytics custom map chart. Custom maps are user-defined maps that are uploaded to Analytics and are used just as standard maps are. Custom maps are accessed in Analytics from the list of maps available with the map chart type.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

EclairGeoData components have the suffix `geodata` and are stored in the `eclair` folder.

Version

EclairGeoData components are available in API version 39.0 and later.

Fields

Field Name	Field Type	Description
maps	EclairMap[]	A list of EclairMap objects. Each EclairMap object specifies the bounding box (if any) and the map name that appears in the user interface.
masterLabel	string	Required. Label for this object. This display value is the internal label that is not translated.

EclairMap

Field Name	Field Type	Description
boundingBoxBottom	double	When bounding-box coordinates are used, this contains the bottom coordinate.
boundingBoxLeft	double	When bounding-box coordinates are used, this contains the left side coordinate.
boundingBoxRight	double	When bounding-box coordinates are used, this contains the right side coordinate.
boundingBoxTop	double	When bounding-box coordinates are used, this contains the top coordinate.
mapLabel	string	Required. The user-interface name of the map. This name appears in the maps list for the map chart in Analytics.
mapName	string	Required. Label for this object. This display value is the internal label that is not translated.
projection	string	Required. The type of map projection used to create the map. Valid values are: • Equirectangular • Mercator • AlbersUSA

Declarative Metadata Sample Definition

The following is an example of an EclairGeoData component:

```
          <EclairGeoData xmlns="http://soap.sforce.com/2006/04/metadata"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
            <content xsi:nil="true"/>
            <maps>
              <boundingBoxBottom>0.0</boundingBoxBottom>
              <boundingBoxLeft>100.0</boundingBoxLeft>
              <boundingBoxRight>100.0</boundingBoxRight>
              <boundingBoxTop>0.0</boundingBoxTop>
              <mapLabel>WorldMap0 Label</mapLabel>
              <mapName>WorldMap0</mapName>
              <projection>Equirectangular</projection>
```

```
    </maps>
    <maps>
      <boundingBoxBottom>1.0</boundingBoxBottom>
      <boundingBoxLeft>101.0</boundingBoxLeft>
      <boundingBoxRight>101.0</boundingBoxRight>
      <boundingBoxTop>1.0</boundingBoxTop>
      <mapLabel>WorldMap1 Label</mapLabel>
      <mapName>WorldMap1</mapName>
      <projection>Mercator</projection>
    </maps>
    <masterLabel>WorldMapGeoDataToCreate Label</masterLabel>
  </EclairGeoData>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
  <types>
    <members>*</members>
    <name>EclairGeoData</name>
  </types>
  <version>39.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

EmailServicesFunction

Represents an email service. This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

EmailServicesFunction components have the suffix `.xml` and are stored in the `emailservices` folder.

Version

EmailServicesFunction components are available in API version 42.0 and later.

Fields

Field Name	Field Type	Description
`apexClass`	string	Required. The name of the Apex class that the email service uses to process inbound messages.
`attachmentOption`	EmailServicesAttOptions (enumeration of type string)	Required. Indicates the types of attachments the email service accepts. One of the following values: • `None`—The email service accepts the message but discards any attachment. • `NoContent`—The attachment metadata (filename, MIME type, and so on) is provided to the Apex class, but the body is set to `null`. • `TextOnly`—The email service only accepts the following types of attachments: – Attachments with a Multipurpose Internet Mail Extension (MIME) type of text. – Attachments with a MIME type of application/octet-stream and a file name that ends with either a .vcf or .vcs extension. These are saved as text/x-vcard and text/calendar MIME types, respectively. • `BinaryOnly`—The email service only accepts binary attachments, such as image, audio, application, and video files. • `All`—The email service accepts any type of attachment.
`authenticationFailureAction`	EmailServicesErrorAction (enumeration of type string)	Required. Indicates what the email service does with messages that fail or do not support any of the authentication protocols if the `isAuthenticationRequired` field is true. One of the following values: • `UseSystemDefault`—The system default is used. • `Bounce`—The email service returns the message to the sender with a notification that explains why the message was rejected. • `Discard`—The email service deletes the message without notifying the sender. • `Requeue`—The email service queues the message for processing in the next 24 hours. If the message is not processed within 24 hours, the email service returns the message to the sender with a notification that explains why the message was rejected.
`authorizationFailureAction`	EmailServicesErrorAction (enumeration of type string)	Required. Indicates what the email service does with messages received from senders who are not listed in the `authorizedSenders` field on either the email service or email service address. One of the following values: • `UseSystemDefault`—The system default is used.

Field Name	Field Type	Description
		• `Bounce`—The email service returns the message to the sender with a notification that explains why the message was rejected. • `Discard`—The email service deletes the message without notifying the sender. • `Requeue`—The email service queues the message for processing in the next 24 hours. If the message is not processed within 24 hours, the email service returns the message to the sender with a notification that explains why the message was rejected.
`authorizedSenders`	string	Configures the email service to only accept messages from the email addresses or domains listed in this field. If the email service receives a message from an unlisted email address or domain, the email service performs the action specified in the `authorizationFailureAction` field. Leave this field blank if you want the email service to receive email from any email address.
`emailServicesAddresses`	EmailServicesAddress[]	A list of EmailServiceAddress records.
`errorRoutingAddress`	email	The destination email address for error notification email messages when `isErrorRoutingEnabled` is `true`.
`functionInactiveAction`	EmailServicesErrorAction (enumeration of type string)	Required. Indicates what the email service does with messages it receives when the email service itself is inactive. One of the following values: • `UseSystemDefault`—The system default is used. • `Bounce`—The email service returns the message to the sender with a notification that explains why the message was rejected. • `Discard`—The email service deletes the message without notifying the sender. • `Requeue`—The email service queues the message for processing in the next 24 hours. If the message is not processed within 24 hours, the email service returns the message to the sender with a notification that explains why the message was rejected.
`functionName`	string	Required. The name of the email service in the API. This name can contain only underscores and alphanumeric characters and must be unique in your org. The value in this 64-character field must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. In managed packages, this field prevents naming conflicts on package installations. This field is automatically generated, but you can supply your own value if you create the record using the API. With this field, a developer can change the object's name in a managed package and the changes are reflected in a subscriber's organization. Note: When creating large sets of data, always specify a unique `functionName` for each record. If no `functionName` is

Field Name	Field Type	Description
		specified, performance may slow while Salesforce generates one for each record.
`isActive`	boolean	Indicates whether this object is active (`true`) or not (`false`).
`isAuthenticationRequired`	boolean	Configures the email service to verify the legitimacy of the sending server before processing a message. The email service uses the SPF, SenderId, and DomainKeys protocols to verify the sender's legitimacy: If the sending server passes at least one of these protocols and does not fail any, the email service accepts the email. If the server fails a protocol or does not support any of the protocols, the email service performs the action specified in the `authenticationFailureAction` field.
`isErrorRoutingEnabled`	boolean	When incoming email messages can't be processed, indicates whether error notification email messages are routed to a chosen address or to the senders.
`isTextAttachmentsAsBinary`	boolean	If `true`, text attachments are supplied to the Apex code as a `Messaging.BinaryAttachment` instead of as a `Messaging.TextAttachment`. This means that the body is supplied as an Apex Blob instead of as an Apex String.
`isTlsRequired`	boolean	Not currently in use.
`overLimitAction`	EmailServicesErrorAction (enumeration of type string)	Required. Indicates what the email service does with messages if the total number of messages processed by all email services combined has reached the daily limit for your organization. One of the following values: • `UseSystemDefault`—The system default is used. • `Bounce`—The email service returns the message to the sender with a notification that explains why the message was rejected. • `Discard`—The email service deletes the message without notifying the sender. • `Requeue`—The email service queues the message for processing in the next 24 hours. If the message is not processed within 24 hours, the email service returns the message to the sender with a notification that explains why the message was rejected. The system calculates the limit by multiplying the number of user licenses by 1,000.

EmailServicesAddress

Each email service has one or more email addresses to which users can send messages for processing. An email service only processes messages it receives at one of its addresses.

Field Name	Field Type	Description
`authorizedSenders`	string	Configures the email service address to only accept messages from the email addresses or domains listed in this field. If the email service address receives a message from an unlisted email address or domain, the email service performs the action specified in the `authorizationFailureAction` field of its associated email service. Leave this field blank if you want the email service address to receive email from any email address.
`developerName`	string	Required. The name of the object in the API. This name can contain only underscores and alphanumeric characters and must be unique in your org. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. This 25-character field must be unique among other EmailServicesAddress records under the same EmailServiceFunction parent. In managed packages, this field prevents naming conflicts on package installations. This field is automatically generated, but you can supply your own value if you create the record using the API. With this field, a developer can change the object's name in a managed package and the changes are reflected in a subscriber's organization. Note: When creating large sets of data, always specify a unique `developerName` for each record. If no `developerName` is specified, performance might be slow while Salesforce generates one for each record.
`isActive`	boolean	Indicates whether this object is active (`true`) or not (`false`).
`localPart`	string	Required. The local-part of the email service address, which is the string that comes before the @ symbol. For the local-part of a Salesforce email address, all alphanumeric characters are valid, plus the following special characters: `! # $ % & amp; ' * / = ? ^ _ + - ` { | } ~ ,` The dot character (.) is also valid as long as it's not the first or last character. Email addresses aren't case sensitive.
`runAsUser`	string	Required. The username of the user whose permissions the email service assumes when processing messages sent to this address.

Wildcard Support in the Manifest File

This metadata type doesn't support the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

EmailTemplate

Represents a template for an email, mass email, list email, or Sales Engagement email. Supported in first-generation managed packages only.

This type extends the MetadataWithContent metadata type and inherits its `content` and `fullName` fields.

Note: First-generation packaging only is supported for Lightning email templates.

File Suffix and Directory Location

The file suffix is `.email` for the template file. The accompanying metadata file is named *`EmailTemplateName`*`-meta.xml`.

EmailTemplate components are stored in the `email` folder in the corresponding package directory. For example, for an email template named SampleTemplate in the sampleFolder folder, there's a `SampleTemplate-meta.xml` in the `email/sampleFolder` of the package.

Retrieving Email Templates

You can't use the wildcard (*) symbol with email templates in `package.xml`. To retrieve the list of email templates for populating `package.xml` with explicit names, call `listMetadata()` and pass in `EmailTemplate` as the type.

The following example shows folders in `package.xml`:

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>MyDBFolder/MyDBName</members>
        <name>Dashboard</name>
    </types>
    <types>
        <members>MyDocumentFolder/MyDocumentName</members>
        <name>Document</name>
    </types>
    <types>
        <members>unfiled$public/MarketingProductInquiryResponse</members>
        <members>unfiled$public/SalesNewCustomerEmail</members>
        <name>EmailTemplate</name>
    </types>
    <types>
        <members>MyReportFolder/MyReportName</members>
        <name>Report</name>
    </types>
    <version>64.0</version>
</Package>
```

Version

Email templates are available in API version 12.0 and later.

Fields

This metadata type contains the following fields:

Field Name	Field Type	Description
`apiVersion`	double	The API version if it's a Visualforce email template. Every Visualforce email template has an API version specified at creation. This field is available in API version 16.0 and later.
`attachedDocuments`	string[]	A list of references to documents in your organization. These documents are included as attachments in the email template. Each document is referenced by its path, for example `MyFolder/MyDocument.txt`.
`attachments`	Attachment[]	A list of attachments for the email template.
`available`	boolean	Required. Indicates whether this template is offered to users when sending an email (`true`) or not (`false`).
`content`	base64Binary	Content of the email template. Base 64-encoded binary data. Before making an API call, client applications must encode the binary attachment data as base64. Upon receiving a response, client applications must decode the base64 data to binary. This conversion is handled for you by a SOAP client. This field contains: • Binary content of the email body if `type` is set to `text` • HTML email content if `type` is set to `html` • HTML body if `type` is set to `custom` • Visualforce body if `type` is set to `visualforce` This field is inherited from the MetadataWithContent component.
`description`	string	The email template description describes the reason for creating the template.
`encodingKey`	Encoding (enumeration of type string)	Required for Classic email templates. The default encoding setting is Unicode: `UTF-8`. Change it if your template requires data in a different format. Valid values include: • `UTF-8`—Unicode (UTF-8) • `ISO-8859-1`—General US & Western Europe (ISO-8859–1, ISO-LATIN-1) • `Shift_JIS`—Japanese (Shift-JIS) • `ISO-2022-JP`—Japanese (JIS) • `EUC-JP`—Japanese (EUC-JP) • `x-SJIS_0213`—Japanese (Shift-JIS_2004) • `ks_c_5601-1987`—Korean (ks_c_5601-1987) • `Big5`—Traditional Chinese (Big5) • `GB2312`—Simplified Chinese (GB2312) • `Big5-HKSCS`—Traditional Chinese Hong Kong (Big5–HKSCS) Lightning email templates don't use this field. Instead, the encoding values are taken directly from the user's encoding settings.
`fullName`	string	The email template developer name used as a unique identifier for API access. The `fullName` can contain only underscores and alphanumeric characters. It must be unique, begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. If this field contained

Field Name	Field Type	Description
		characters before version 14.0 that are no longer allowed, the characters were stripped out of this field, and the previous value of the field was saved in the `name` field. This field is inherited from the Metadata component.
`letterhead`	string	The letterhead name associated with this email template. Only available when `type` is set to `html`.
`name`	string	Required. Email template name. The list of characters allowed in the `fullName` field has been reduced for versions 14.0 and later. This field contains the value contained in the `fullName` field before version 14.0.
`packageVersions`	PackageVersion[]	The list of package versions for any managed packages containing components that are referenced by this email template. This field is only relevant for Visualforce email templates. For more information about managed packages, see Second-Generation Managed Packages in the *Salesforce DX Developer Guide*. This field is available in API version 16.0 and later.
`relatedEntityType`	Object Name (enumeration of type string)	Reserved for future use with Lightning Experience.
`style`	EmailTemplateStyle (enumeration of type string)	Required. The style of the template. This field is only available when `type` is set to `html`. Valid style values include: • `none` • `freeForm` • `formalLetter` • `promotionRight` • `promotionLeft` • `newsletter` • `products`
`subject`	string	The email subject. The limit is 1,000 characters for Lightning email templates and 230 characters for Classic email templates.
`textOnly`	string	The text of the email body if `type` is set to `html` or `custom`.
`type`	EmailTemplateType (enumeration of type string)	Required. The email template type. The valid values are: • `text` - all users can create or change text email templates. • `html` - administrators and users with the "Edit HTML Templates" permission can create HTML email templates based on a letterhead. • `custom` - administrators and users with the "Edit HTML Templates" permission can create custom HTML email templates without using a

Field Name	Field Type	Description
		letterhead. You must either know HTML or obtain the HTML code to insert in your email template. • `visualforce` - administrators and users with the Customize Application permission can create email templates using Visualforce.
`UiType`	EmailTemplateUiType (enumeration of type string)	Indicates the user interface where this template is usable. Valid values are: • `Aloha` (Salesforce Classic) • `SFX` (Lightning Experience) • `SFX_Sample` (Lightning Experience Sample) If `UiType` is `SFX`, the `type` must be `custom`. Packaging is supported for Salesforce Classic email templates only.

Example:

```
<EmailTemplate>
    <available>true</available>
    <description>Notification that user has been added to a community.</description>
    <encodingKey>UTF-8</encodingKey>
    <name>Communities: New Member Welcome Email</name>
    <style>none</style>
    <subject>Welcome to {!Community_Name}</subject>
    <type>custom</type>
    <uiType>Aloha</uiType>
</EmailTemplate>
```

Attachment

Attachment represents an email attachment.

Field	Field Type	Description
`content`	base64Binary	Required. The attachment content. Base 64-encoded binary data. Before making an API call, client applications must encode the binary attachment data as base64. Upon receiving a response, client applications must decode the base64 data to binary. This conversion is handled for you by a SOAP client.
`name`	string	Required. The attachment file name.

Declarative Metadata Sample Definition

Here's a sample XML definition of an email template.

```
<?xml version="1.0" encoding="UTF-8"?>
<EmailTemplate xmlns="http://soap.sforce.com/2006/04/metadata">
    <available>true</available>
    <description>Sample Email Template</description>
    <encodingKey>ISO-8859-1</encodingKey>
    <name>Sample Email Template</name>
    <style>none</style>
    <subject>Sample email subject</subject>
    <textOnly>Your case has been resolved.</textOnly>
    <type>custom</type>
</EmailTemplate>
```

Wildcard Support in the Manifest File

This metadata type doesn't support the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SEE ALSO:

Letterhead

EmbeddedServiceBranding

Represents the branding for each Embedded Service deployment. This type extends the Metadata metadata type and inherits its `fullName` field.

This object works only with the legacy chat products. For Messaging for In-app and Web, use the BrandingSet object.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

EmbeddedServiceBranding components are stored in the *developer_name*`.EmbeddedServiceBranding` file in the `EmbeddedServiceBranding` folder.

Version

EmbeddedServiceBranding is available in API version 39.0 and later.

Fields

Field Name	Field Type	Description
contrastInvertedColor	string	Accent branding color used in the embedded component, displayed as a hexadecimal value. Changes made to this field in the API aren't reflected in the embedded component.
contrastPrimaryColor	string	Accent branding color used in the embedded component, displayed as a hexadecimal value.
embeddedServiceConfig	string	Required. The Embedded Service configuration that this branding applies to.
font	string	Font used in the text of the embedded component.
height	int	Height of the embedded component. Available in API version 43.0 and later.
masterLabel	string	Required. The name of the Embedded Service configuration node.
navBarColor	string	Color used for the header in the embedded component, displayed as a hexadecimal value.
navBarTextColor	string	Color used for the text and icons in the header in the embedded component, displayed as a hexadecimal value. Available in API version 49.0 and later.
primaryColor	string	Primary branding color used in the embedded component, displayed as a hexadecimal value.
secondaryColor	string	Secondary branding color used in the embedded component, displayed as a hexadecimal value.
secondaryNavBarColor	string	Secondary branding color used for the header in the embedded component, displayed as a hexadecimal value. It applies to the header in the chat feature when it's trying to reconnect because of lost internet connection. Available in API version 49.0 and later.
width	int	Width of the embedded component. Available in API version 43.0 and later.

Declarative Metadata Sample Definition

The following is an example of an EmbeddedServiceBranding file.

```
<?xml version="1.0" encoding="UTF-8"?>
<EmbeddedServiceBranding xmlns="http://soap.sforce.com/2006/04/metadata">
    <contrastInvertedColor>#ffffff</contrastInvertedColor>
    <contrastPrimaryColor>#333333</contrastPrimaryColor>
    <embeddedServiceConfig>EswConfig001</embeddedServiceConfig>
    <font>Salesforce Sans</font>
    <height>498</height>
    <masterLabel>EmbeddedServiceBranding_Parent04IRM000000002a_16033cd2c16</masterLabel>
```

```
    <navBarColor>#222222</navBarColor>
    <primaryColor>#222222</primaryColor>
    <secondaryColor>#005290</secondaryColor>
    <width>320</width>
</EmbeddedServiceBranding>
```

Wildcard Support in the Manifest File

This metadata type doesn't support the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

EmbeddedServiceConfig

Represents a setup node for creating an Embedded Service for Web deployment. This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

EmbeddedServiceConfig components have the suffix `.EmbeddedServiceConfig` and are stored in the `EmbeddedServiceConfig` folder.

Version

EmbeddedServiceConfig is available in API version 37.0 and later.

Fields

Field Name	Field Type	Description
`areGuestUsersAllowed`	boolean	Specifies whether a user must be logged in to access an embedded component. Available in API version 45.0 and later.
`authMethod`	EmbeddedServiceAuthMethod (enumeration of type string)	Type of login method selected for this Embedded Service deployment. Valid values are: • `CommunitiesLogin`–Customers log in using Communities. • `CustomLogin`–Customers log in using your own custom authentication. Available in API version 43.0 and later.

Field Name	Field Type	Description
branding	string	The branding set that has all of the branding configurations for this Embedded Service configuration. Available in API version 52.0 and later.
customMinimizedComponent	string	The custom Lightning component that's used in this Embedded Service deployment in its minimized state. Available in API version 43.0 to 45.0.
deploymentFeature	EmbeddedServiceDeploymentFeature (enumeration of type string)	The conversation type of this Embedded Service deployment. Valid values are: • EmbeddedMessaging—Messaging for In-App and Messaging for Web deployments • Flows • FieldService • LiveAgent • None Available in API version 52.0 and later.
deploymentType	EmbeddedServiceDeploymentType (enumeration of type string)	The platform this Embedded Service is deployed to. Valid values are: • Mobile—For future use • Web • API Available in API version 51.0 and later.
embeddedServiceAppointmentSettings	EmbeddedServiceAppointmentSettings[]	The settings of the Embedded Service deployment whose deploymentFeature is FieldService. Available in API version 46.0 and later.
embeddedServiceCustomComponents	EmbeddedServiceCustomComponent on page 918[]	The custom components used in this Embedded Service deployment. Available in API version 44.0 and later.
embeddedServiceCustomLabels	EmbeddedServiceCustomLabel on page 919[]	The custom labels used in this Embedded Service deployment. Available in API version 44.0 and later.
embeddedServiceCustomizations	EmbeddedServiceCustomization on page 919[]	The customizations used in this Embedded Service deployment. Each customization is associated with a static resource. Available in API version 50.0 and later.
embeddedServiceFlowConfig	EmbeddedServiceFlowConfig on page 923[]	Represents a setup node for creating an embedded flow. Available in API version 45.0 and later.
embeddedServiceFlows	EmbeddedServiceFlow on page 922[]	All of the flows used by this Embedded Service deployment. Available in API version 45.0 and later.
embeddedServiceLayouts	EmbeddedServiceLayout[]	The layout of an Appointment Management deployment of an Embedded Service. Available in API version 44.0 and later.
isEnabled	boolean	Indicates if this Embedded Service deployment is enabled (true).
isTermsAndConditionsEnabled	boolean	Indicates whether Terms and Conditions is displayed. Displaying Terms and Conditions is supported if the deploymentFeature is either

Field Name	Field Type	Description
		`EmbeddedMesssaging` or `LiveAgent`. The default is `false`. Available in API version 59.0 and later.
`isTermsAndConditionsRequired`	boolean	Indicates whether acceptance of the Terms and Conditions is required before starting a chat. Displaying Terms and Conditions is supported if the `deploymentFeature` is either `EmbeddedMesssaging` or `LiveAgent`. The default is `false`. Available in API version 59.0 and later.
`masterLabel`	string	Required. The name of the Embedded Service configuration node. Available in API version 37.0 and later.
`shouldHideAuthDialog`	boolean	Specifies whether the prompt that the customer log in again during a flow is hidden (`true`) or not (`false`). When it's hidden, the customer is taken directly to your login page. This field is set to `false` by default. Available in API version 43.0 and later.
`site`	string	Required. The name of the Experience site or website connected to this Embedded Service deployment. Available in API version 37.0 and later.

EmbeddedServiceAppointmentSettings

Returns the settings of an Embedded Service deployment whose `deploymentFeature` is `FieldService`. Available in API version 46.0 and later.

Field Name	Description
`appointmentConfirmImg`	**Field Type** string **Description** The URL of the image to display when an appointment is confirmed.
`enabled`	**Field Type** boolean **Description** Required. Indicates whether this deployment is enabled. The default is `false`.
`homeImg`	**Field Type** string **Description** The URL of the image to display on the appointment management widget home screen.
`logoImg`	**Field Type** string

Field Name	Description
	Description The URL of the logo to display in the appointment management widget.
`shouldShowExistingAppointment`	**Field Type** boolean **Description** Indicates whether existing appointments are displayed in the appointment management widget. The default is `false`.
`shouldShowNewAppointment`	**Field Type** boolean **Description** Indicates whether new appointments are displayed in the appointment management widget. The default is `false`.

EmbeddedServiceCustomComponent

Returns a custom component that's associated with an EmbeddedServiceConfig setup.

Field Name	Field Type	Description
`customComponent`	string	The name of the custom component.
`customComponentType`	EmbeddedServiceCustomComponentType (enumeration of type string)	The type of custom component. Valid values are: • `LA_Prechat` (component for pre-chat in Embedded Chat) • `LA_Minimized` (component for the minimized chat window) • `LA_PlainTextChatMessage` (component for the text area in Embedded Chat) • `MIAW_Prechat`(component for the prechat form in Messaging for In-App and Web) • `MIAW_Header`(component for the header of the messaging conversation window in Messaging for In-App and Web) • `MIAW_TextMessage`(component for the text message bubbles in a Messaging for In-App and Web conversation)

EmbeddedServiceCustomLabel

Returns a custom label that's associated with an EmbeddedServiceConfig setup.

Field Name	Field Type	Description
`customLabel`	string	The customized label that appears in the embedded component.
`feature`	EmbeddedServiceFeature (enumeration of type string)	The feature that this embedded component uses. Valid values are: • `Base` • `ChannelMenu` • `EmbeddedMessaging`—Messaging for In-App and Messaging for Web deployments • `FieldService` • `Flows` • `LiveAgent` • `NotInUse`
`labelKey`	EmbeddedServiceLabelKey (enumeration of type string)	The type of label for this embedded component. The value corresponds to the label within a label group (substate of chat state or page type).

EmbeddedServiceCustomization

Returns the customization associated with the Embedded Service feature. Available in API version 50.0 and later.

Field Name	Field Type	Description
`customizationName`	string	Required. The name of the customization applied to the embedded service. This name can contain only underscores and alphanumeric characters and must be unique in an EmbeddedServiceConfig setup. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
`description`	string	A description of the customization.
`embeddedServiceResources`	EmbeddedServiceResource on page 922[]	Required. The reference to the static resource that contains the javascript file of the customization.

EmbeddedServiceForm

Returns the form that's used for pre-chat. Available in API version 62 and later.

Field Name	Field Type	Description
displayContext	EmbeddedServiceFormDisplayContext (enumeration of type string)	Required. Indicates when we display this form. Values are: • `Session`- The pre-chat form is displayed every session. • `Conversation`- Tthe pre-chat form is displayed every conversation. • `None`- Don't select this option.
isActive	Boolean	Indicates whether the form is active. The default value is false.

EmbeddedServiceFormField

Represents an individual field in a prechat form. Available in API version 62 or higher.

Field Name	Field Type	Description
formField	string or enum	Required. Maps to either a StandardMessingChannelParameter or a CustomMessagingChannelParameter
messagingChannelParameterType	MessagingChannelParameterType (enumeration of type string) OR StandardMessagingChannelParameter (enumeration of type string)	Required. Indicates whether `formField` points to a StandardMessagingChannelParameter or a CustomMessagingChannelParameter. Please match the Enum value here to the formField passed in. Valid values for MessagingChannelParameterType are: • `Standard` - select this field when you want to reference a StandardMessagingChannelParameter • `Custom` - select this field when you want to reference a CustomMessagingChannelParameter Valid values for StandardMessagingChannelParameter are: • `FirstName` • `LastName` • `Email` • `Subject`

Field Name	Field Type	Description
formFieldType	EmbeddedServiceFormFieldType (enumeration of type string)	Required. Indicates the type of data is passed by the End User when filling out the Prechat form. Valid values are: • Text • Email • Phone • Number • Checkbox • Choicelist
displayOrder	Integer	Indicates the relative order of the form field in the prechat form. For Prechat fields where `isHidden` is set to `true` displayOrder must be set to -1. For Prechat fields where `isHidden` is set to `false`, this value must be greater than or equal to 0.
isRequired	Boolean	Indicates whether a Prechat form field is required. If `isHidden` is set to `true`, `isRequired` must be set to `false`.
isHidden	Boolean	Indicates whether a Prechat form field is displayed in the UI. `isHidden` can only be set to `true` if the `formField` field is set to `CustomMessagingChannelParameter` and `messagingChannelParameterType` is set to `Custom`.
choiceList	ChoiceList	The choice list that's associated with `EmbeddedServiceFormField`. ChoiceList can only be associated if the formField is set to CustomMessagingChannelParameter and messagingChannelParameterType is set to Custom.
embeddedServiceCustomLabels	EmbeddedServiceCustomLabel[]	The custom label for this field.

embeddedServiceMessagingChannel

Returns the settings of an Embedded Service deployment whose

```
deploymentFeature
```

is EmbeddedMessaging. Available in API version 62 or higher.

Field Name	Field Type	Description
businessHours	String	The Business Hours record that's associated with this specific embedded service deployment.
isEnabled	Boolean	Required. Indicates whether this embedded service deployment is enabled.
messagingChannel	String	Required. The MessagingChannel record that's associated with this specific embedded service deployment.
shouldShowDeliveryReceipts	Boolean	Required. Indicates whether delivery receipts are shown in the UI.
shouldShowEmojiSelection	Boolean	Required. Indicates whether emoji selection is shown in the UI.
shouldShowReadReceipts	Boolean	Required. Indicates whether read reciepts are shown in the UI.
shouldShowTypingIndicators	Boolean	Required. Indicates whether typing indicators are shown in the UI.
shouldStartNewLineOnEnter	Boolean	Required. Indicates whether pressing Enter starts a new line of text in the UI.

EmbeddedServiceResource

Returns the static resource associated with the Embedded Service Chat feature customization. Available in API version 50.0 and later.

Field Name	Field Type	Description
`resource`	string	Required. The ID of the static resource that contains the javascript file of the customization.
`resourceType`	EmbeddedServiceResourceType (enumeration of type string)	Required. The embedded service feature to customize. Valid values are: • `ChatInvitation`—Use for Chat deployments. • `SettingsFile`—Use if you're configuring a settings file for a Channel Menu deployment.

EmbeddedServiceFlow

Returns an embedded flow that's associated with an EmbeddedServiceConfig setup.

Field Name	Field Type	Description
`flow`	string	The developer name of the flow.
`flowType`	EmbeddedServiceFlowType(enumeration of type string)	The type of flow. Valid values are: • `FS_CancelAppointment` • `FS_Flow` • `FS_ModifyAppointment` • `FS_NewAppointment` • `LA_Survey`
`isAuthenticationRequired`	boolean	Indicates whether users are required to log in to access the Embedded Service component. The value can't be `true` for the `FS_Flow` value and must be `true` for all other values.

EmbeddedServiceFlowConfig

Returns the EmbeddedServiceFlowConfig type.

Field Name	Field Type	Description
`enabled`	boolean	Indicates whether the embedded flow is enabled.

EmbeddedServiceLayout

Returns the layout of an Embedded Service deployment whose `deploymentFeature` is `FieldService`. Available in API version 44.0 and later.

Field Name	Field Type	Description
`embeddedServiceLayoutRules`		The appointment statuses that the layout of the Embedded Service deployment is valid for.
`layout`		The FlexiPage that represents the layout of this Embedded Service deployment.
`layoutType`		The type of layout applied to the Embedded Service deployment. Values are: • `FS_AppointmentHome`

EmbeddedServiceLayoutRule

Returns an appointment status for which the Embedded Service layout is valid for. This subtype is for Embedded Service deployments whose `deploymentFeature` is `FieldService`. Available in API version 44.0 and later.

Field Name	Field Type	Description
`appointmentStatus`	string	The service appointment status that the EmbeddedServiceLayout subtype is valid for.

Declarative Metadata Sample Definition

The following is an example of an EmbeddedServiceConfig file.

```
<?xml version="1.0" encoding="UTF-8"?>
<EmbeddedServiceConfig xmlns="http://soap.sforce.com/2006/04/metadata">
    <areGuestUsersAllowed>false</areGuestUsersAllowed>
    <deploymentType>Mobile</deploymentType>
    <deploymentFeature>EmbeddedMessaging</deploymentFeature>
    <masterLabel>ESWOne</masterLabel>
    <shouldHideAuthDialog>false</shouldHideAuthDialog>

    <embeddedServiceMessagingChannel>
        <isEnabled>true</isEnabled>
        <shouldShowTypingIndicators>false</shouldShowTypingIndicators>
        <shouldShowReadReceipts>false</shouldShowReadReceipts>
        <shouldShowDeliveryReceipts>false</shouldShowDeliveryReceipts>
        <shouldShowEmojiSelection>false</shouldShowEmojiSelection>
        <shouldStartNewLineOnEnter>false</shouldStartNewLineOnEnter>
        <messagingChannel>EM1</messagingChannel>
    </embeddedServiceMessagingChannel>

    <embeddedServiceForms>
        <isActive>true</isActive>
        <displayContext>Session</displayContext>

        <embeddedServiceFormFields>
            <displayOrder>0</displayOrder>
            <formField>_FirstName</formField>
            <messagingChannelParameterType>Standard</messagingChannelParameterType>
            <formFieldType>Text</formFieldType>
            <isHidden>false</isHidden>
            <isRequired>true</isRequired>
            <embeddedServiceCustomLabels>

<customLabel>EM_PreChat_Base_PrechatCustomFieldLabel_133xx0000004GG2_5523048</customLabel>

                <labelKey>EM_PreChat_Base_PrechatCustomFieldLabel</labelKey>
                <feature>EmbeddedMessaging</feature>
            </embeddedServiceCustomLabels>
        </embeddedServiceFormFields>
        <embeddedServiceFormFields>
            <displayOrder>1</displayOrder>
```

```
            <formField>_LastName</formField>
            <messagingChannelParameterType>Standard</messagingChannelParameterType>
            <formFieldType>Text</formFieldType>
            <isHidden>false</isHidden>
            <isRequired>true</isRequired>
            <embeddedServiceCustomLabels>

<customLabel>EM_PreChat_Base_PrechatCustomFieldLabel_133xx0000004GG2_5523058</customLabel>

                <labelKey>EM_PreChat_Base_PrechatCustomFieldLabel</labelKey>
                <feature>EmbeddedMessaging</feature>
            </embeddedServiceCustomLabels>
        </embeddedServiceFormFields>
        <embeddedServiceFormFields>
            <displayOrder>2</displayOrder>
            <formField>FavoriteFood_name</formField>
            <messagingChannelParameterType>Custom</messagingChannelParameterType>
            <formFieldType>ChoiceList</formFieldType>
            <isHidden>false</isHidden>
            <isRequired>true</isRequired>
            <choiceList>Food</choiceList>
        </embeddedServiceFormFields>
    </embeddedServiceForms>
</EmbeddedServiceConfig>
```

Wildcard Support in the Manifest File

This metadata type doesn't support the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

EmbeddedServiceFieldService

Represents a setup node for creating an embedded Appointment Management deployment. This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

EmbeddedServiceFieldService components are stored in a *developer_name*`.EmbeddedServiceFieldService` file in the `EmbeddedServiceFieldService` folder.

Version

EmbeddedServiceFieldService is available in API version 43.0 and later.

Fields

Field Name	Field Type	Description
appointmentBookingFlowName	string	Name of the appointment booking flow for this embedded Appointment Management (beta) deployment.
cancelApptBookingFlowName	string	Name of the appointment cancellation flow for this embedded Appointment Management (beta) deployment.
embeddedServiceConfig	string	Required. The name of the Embedded Service configuration node.
enabled	boolean	Required. Indicates whether this embedded Appointment Management deployment is enabled (`true`).
fieldServiceConfirmCardImg	string	URL of the image used for the confirmation card in embedded Appointment Management (beta).
fieldServiceHomeImg	string	URL of the image used for the home screen in embedded Appointment Management (beta).
fieldServiceLogoImg	string	URL of the logo used for the home screen in embedded Appointment Management (beta).
masterLabel	string	Required. Name of the embedded Appointment Management (beta) deployment.
modifyApptBookingFlowName	string	Name of the appointment modification flow for this embedded Appointment Management (beta) deployment.
shouldShowExistingAppointment	boolean	Specifies whether to display a button on the home screen for customers to access their existing appointments (`true`) or not (`false`). This field is `false` by default.
shouldShowNewAppointment	boolean	Specifies whether to display a button on the home screen for customers to create a new appointment (`true`) or not (`false`). This field is `false` by default.

Declarative Metadata Sample Definition

The following is an example of an EmbeddedServiceFieldService file.

```
<?xml version="1.0" encoding="UTF-8"?>
<EmbeddedServiceFieldService xmlns="http://soap.sforce.com/2006/04/metadata">
    <appointmentBookingFlowName>ESW_FS_BookAppt_Main_Flow</appointmentBookingFlowName>
    <cancelApptBookingFlowName>ESW_FS_CancelAppt_Flow</cancelApptBookingFlowName>
    <embeddedServiceConfig>EswFS</embeddedServiceConfig>
    <enabled>true</enabled>

<fieldServiceConfirmCardImg>https://google.com/AppointmentConfirmationImg.png</fieldServiceConfirmCardImg>

    <fieldServiceHomeImg>https://google.com/HeroImg.png</fieldServiceHomeImg>
    <fieldServiceLogoImg>https://google.com/logo.png</fieldServiceLogoImg>
```

```
<masterLabel>EmbeddedServiceFieldService_Parent04IRM000000007p2AA_162d4270834</masterLabel>

    <modifyApptBookingFlowName>ESW_FS_ModifyAppt_Main_Flow</modifyApptBookingFlowName>
    <shouldShowExistingAppointment>true</shouldShowExistingAppointment>
    <shouldShowNewAppointment>true</shouldShowNewAppointment>
</EmbeddedServiceFieldService>
```

Usage

Note: Any changes you make to the image fields override what you've entered in Setup. We recommend setting your image URLs in Setup.

EmbeddedServiceFlowConfig

Represents a setup node for creating an embedded flow. This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

EmbeddedServiceFlowConfig components are stored in the *developer_name*.`EmbeddedServiceFlowConfig` file in the `EmbeddedServiceFlowConfig` folder.

Version

EmbeddedServiceFlowConfig is available in API version 45.0 and later.

Fields

Field Name	Field Type	Description
`enabled`	boolean	Indicates whether the embedded flow is enabled (`true`) or not (`false`). Defaults to `false`.

Declarative Metadata Sample Definition

The following is an example of an EmbeddedServiceFlowConfig file.

```
<?xml version="1.0" encoding="UTF-8"?>
<EmbeddedServiceFlowConfig xmlns="http://soap.sforce.com/2006/04/metadata">
    <enabled>true</enabled>
</EmbeddedServiceFlowConfig>
```

EmbeddedServiceLiveAgent

Represents a setup node for creating an embedded chat deployment. This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

EmbeddedServiceLiveAgent components are stored in the *`developer_name`*`.EmbeddedServiceLiveAgent` file in the `EmbeddedServiceLiveAgent` folder.

Version

EmbeddedServiceLiveAgent is available in API version 38.0 and later.

Fields

Field Name	Field Type	Description
`avatarImg`	string	Avatar image for this embedded chat deployment.
`customPrechatComponent`	string	The custom Lightning Component that's used for the pre-chat page in this embedded chat deployment.
`embeddedServiceConfig`	string	Required. The name of the embedded service configuration node.
`embeddedServiceQuickActions`	EmbeddedServiceQuickAction	The quick action used by the pre-chat form.
`enabled`	boolean	Required. Indicates whether this embedded chat deployment is enabled (`true`).
`fontSize`	EmbeddedServiceFontSize (enumeration of type string)	Required. The font size for the text in the embedded chat window. One of the following values: • `Small` • `Medium` • `Large`
`headerBackgroundImg`	string	Header background image for this embedded chat window. Removed in API version 49.0.
`isOfflineCaseEnabled`	boolean	Indicates whether offline support is enabled for this embedded chat deployment. Available in API version 43.0 and later.
`isQueuePositionEnabled`	boolean	Indicates whether queue position (displaying the chat visitor's place in line while they wait for an agent) is enabled for this embedded chat deployment. Available in API version 43.0 and later.
`liveAgentChatUrl`	string	The rest endpoint for chats.

Field Name	Field Type	Description
`liveAgentContentUrl`	string	The rest endpoint for cChat content.
`liveChatButton`	string	Required. Reference to a chat button created in Chat setup.
`liveChatDeployment`	string	Required. Reference to a deployment created in Chat setup.
`masterLabel`	string	Required. Name of the embedded chat deployment.
`offlineCaseBackgroundImg`	string	Offline support case form background image for this embedded chat window. Available in API version 43.0 and later.
`prechatBackgroundImg`	string	Pre-chat background image for this embedded chat window.
`prechatEnabled`	string	Required. Indicates whether the embedded chat pre-chat form is enabled for this deployment.
`prechatJson`	string	JSON object of all the fields of the selected pre-chat form in Chat setup.
`scenario`	EmbeddedServiceScenario (enumeration of type string)	Required. The scenario for the embedded chat window that determines which objects to relate to the chat. One of the following values: • `Sales` • `Service` • `Basic`
`smallCompanyLogoImg`	string	Company logo image for this embedded chat window.
`waitingStateBackgroundImg`	string	Chat waiting image for this embedded chat window.

EmbeddedServiceQuickAction

Returns a quick action that's associated with an EmbeddedServiceLiveAgent setup. The quick action includes the pre-chat form fields that the embedded chat window displays and shows the order in which the fields are displayed.

Field Name	Field Type	Description
`embeddedServiceLiveAgent`	string	Reference to the embedded chat deployment.
`order`	int	Order in which this quick action appears in the embedded chat pre-chat form.
`quickActionDefinition`	string	Reference to a quick action.
`quickActionType`	EmbeddedServiceQuickActionType (enumeration of type string)	Quick action type. One of the following values: • `Prechat`–Pre-chat • `OfflineCase`–Offline support (Cases) Available in API version 43.0 and later.

Declarative Metadata Sample Definition

The following is an example of an EmbeddedServiceLiveAgent file.

```
<?xml version="1.0" encoding="UTF-8"?>
<EmbeddedServiceLiveAgent xmlns="http://soap.sforce.com/2006/04/metadata">
    <avatarImg>https://google.com/avatar.png</avatarImg>
    <customPrechatComponent>auraCustomPrechat</customPrechatComponent>
    <embeddedServiceConfig>EswConfig001</embeddedServiceConfig>
    <embeddedServiceQuickActions>

<embeddedServiceLiveAgent>EmbeddedServiceLiveAgent_Parent04Ixx0000000001EAA_15ec5bd2971</embeddedServiceLiveAgent>

        <order>1</order>

<quickActionDefinition>Snapins_Contact_QuickAction_08hRM000000001h</quickActionDefinition>

    </embeddedServiceQuickActions>
    <embeddedServiceQuickActions>

<embeddedServiceLiveAgent>EmbeddedServiceLiveAgent_Parent04Ixx0000000001EAA_15ec5bd2971</embeddedServiceLiveAgent>

        <order>1</order>

<quickActionDefinition>Snapins_Case_OfflineCaseQuickAction_08hRM000000001h</quickActionDefinition>

        <quickActionType>OfflineCase</quickActionType>
    </embeddedServiceQuickActions>
    <embeddedServiceQuickActions>

<embeddedServiceLiveAgent>EmbeddedServiceLiveAgent_Parent04Ixx0000000001EAA_15ec5bd2971</embeddedServiceLiveAgent>

        <order>2</order>

<quickActionDefinition>Snapins_Case_QuickAction_08hRM000000001h</quickActionDefinition>
    </embeddedServiceQuickActions>
    <enabled>true</enabled>
    <fontSize>Medium</fontSize>
    <headerBackgroundImg>https://google.com/headerBackgroundImg.png</headerBackgroundIm>
    <isOfflineCaseEnabled>true</isOfflineCaseEnabled>
    <isQueuePositionEnabled>true</isQueuePositionEnabled>
    <liveChatButton>chatButton01</liveChatButton>
    <liveChatDeployment>liveAgentDeployment01</liveChatDeployment>
   <masterLabel>EmbeddedServiceLiveAgent_Parent04Ixx0000000001EAA_15ec5bd2971</masterLabel>


<offlineCaseBackgroundImg>https://google.com/offlineCaseBackgroundImg.png</offlineCaseBackgroundImg>

   <prechatBackgroundImg>https://google.com/prechatBackgroundImg.png</prechatBackgroundImg>

    <prechatEnabled>true</prechatEnabled>
    <scenario>Service</scenario>
    <smallCompanyLogoImg>https://google.com/smallCompanyLogoImg.png</smallCompanyLogoImg>
```

```
<waitingStateBackgroundImg>https://google.com/waitingImage.png</waitingStateBackgroundImg>
</EmbeddedServiceLiveAgent>
```

Usage

EmbeddedServiceLiveAgent represents a Chat configuration that is added to your web page. The EmbeddedServiceLiveAgent record contains a unique combination of a chat button and the Chat deployment that the administrator selects during setup.

To create an EmbeddedServiceLiveAgent record:

1. Create a Chat Deployment record.
2. Create a Chat Button record.
3. Create an EmbeddedServiceConfig record.
4. Set the fields for the Chat Deployment record, Chat Button record, and EmbeddedServiceConfig record as references on the EmbeddedServiceLiveAgent record.

Any changes you make to the image fields override what you've entered in Setup. We recommend setting your image URLs in Setup.

Wildcard Support in the Manifest File

This metadata type doesn't support the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

EmbeddedServiceMenuSettings

Represents a setup node for creating a channel menu deployment. Channel menus list the ways in which customers can contact your business. This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

EmbeddedServiceMenuSettings components are stored in the `developer_name.EmbeddedServiceMenuSettings` folder.

Version

EmbeddedServiceMenuSettings components are available in API version 47.0 and later.

Fields

Field Name	Field Type	Description
`branding`	string	The developer name of the associated BrandingSet.
`embeddedServiceCustomLabels`	EmbeddedServiceCustomLabel[]	Represents a customized label that appears in the embedded component for a particular channel menu deployment.

Field Name	Field Type	Description
`embeddedServiceCustomizations`	EmbeddedServiceCustomization on page 932[]	The customizations used in this Embedded Service deployment. Each customization is associated with a static resource. Available in API version 50.0 and later.
`embeddedServiceMenuItems`	EmbeddedServiceMenuItem[]	Represents a channel menu item that lists a way in which customers can contact your business.
`isEnabled`	boolean	If `true` (default), the deployment is enabled. If `false`, the deployment is disabled.
`masterLabel`	string	Required. The name of the channel menu deployment.
`site`	string	Required. The name of the Experience site or website connected to this channel menu deployment.

EmbeddedServiceCustomLabel

Represents the custom labels used in your channel menu deployment.

Field Name	Field Type	Description
`customLabel`	string	The customized label that appears in the channel menu.
`feature`	EmbeddedServiceFeature (enumeration of type string)	The feature using the custom label. For channel menu deployments, the value is `ChannelMenu`.
`labelKey`	EmbeddedServiceLabelKey (enumeration of type string)	The type of label for this embedded component. The value corresponds to the label within a label group (substate of chat state or page type).

EmbeddedServiceCustomization

Returns the customization associated with the Embedded Service feature. Available in API version 50.0 and later.

Field Name	Field Type	Description
`customizationName`	string	Required. The name of the customization applied to the embedded service. This name can contain only underscores and alphanumeric characters and must be unique in an EmbeddedServiceConfig setup. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
`description`	string	A description of the customization.

Field Name	Field Type	Description
`embeddedServiceResources`	EmbeddedServiceResource on page 933[]	Required. The reference to the static resource that contains the javascript file of the customization.

EmbeddedServiceResource

Returns the static resource associated with the Embedded Service Chat feature customization. Available in API version 50.0 and later.

Field Name	Field Type	Description
`resource`	string	Required. The ID of the static resource that contains the javascript file of the customization.
`resourceType`	EmbeddedServiceResourceType (enumeration of type string)	Required. The embedded service feature to customize. Only the Chat feature is supported. Valid values are: • `ChatInvitation`

EmbeddedServiceMenuItem

Represents an item in a channel menu.

Field Name	Field Type	Description
`channel`	string	The ID of the channel type. If `channelType` is `Phone` or `CustomURL`, this field is `null`.
`channelType`	EmbeddedServiceChannelType (enumeration of type string)	The type of communication channel. Values are: • `EmbeddedMessaging` • `EmbeddedServiceConfig` • `MessagingChannel` • `Phone` • `CustomURL`
`customUrl`	string	A custom URL that appears in the menu. The `shouldOpenUrlInSameTab` field determines where the URL opens.
`displayOrder`	int	The item's order in the menu, such as 1 or 2.
`embeddedServiceCustomLabels`	EmbeddedServiceCustomLabel[]	Represents the custom labels used in your channel menu item.

Field Name	Field Type	Description
iconUrl	string	The icon URL for the menu item. Icons can be used only for phone, SMS, custom URL, and chat menu items.
isDisplayedOnPageLoad	boolean	If true, the menu item is displayed on page load. Available in API version 49.0 and later.
itemName	string	A unique custom name for the menu item, which is visible in the user interface.
osOptionsHideInIOS	boolean	If true, the menu item is hidden in iOS.
osOptionsHideInLinuxOS	boolean	If true, the menu item is hidden in Linux operating system.
osOptionsHideInMacOS	boolean	If true, the menu item is hidden in Mac operating system.
osOptionsHideInOtherOS	boolean	If true, the menu item is hidden in any operating system other than iOS, Linux, Mac, and Windows.
osOptionsHideInWindowsOS	boolean	If true, the menu item is hidden in Windows operating system.
phoneNumber	string	The phone number for menu items whose channelType is Phone.
shouldOpenUrlInSameTab	boolean	If the menu item's channelType is CustomURL, this field indicates whether the link opens in the same tab (true) or a new tab (false).

Declarative Metadata Sample Definition

The following is an example of an EmbeddedServiceMenuSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<EmbeddedServiceMenuSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <embeddedServiceCustomLabels>

<customLabel>CM_Container_Header_Primary_Greeting_3MsRM0000004CB5_6181150</customLabel>
        <labelKey>CM_Container_Header_Primary_Greeting</labelKey>
    </embeddedServiceCustomLabels>
    <embeddedServiceCustomLabels>

<customLabel>CM_Container_Header_Secondary_Greeting_3MsRM0000004CB5_4637097</customLabel>

        <labelKey>CM_Container_Header_Secondary_Greeting</labelKey>
    </embeddedServiceCustomLabels>
    <embeddedServiceMenuItems>
        <channel>Chat</channel>
        <channelType>EmbeddedServiceConfig</channelType>
        <displayOrder>1</displayOrder>
```

```
        <embeddedServiceCustomLabels>

<customLabel>CM_Container_MenuItems_WebChatUnavailable_3miRM0000004CuZ_8003848</customLabel>

            <labelKey>CM_Container_MenuItems_WebChatUnavailable</labelKey>
        </embeddedServiceCustomLabels>
        <embeddedServiceCustomLabels>

<customLabel>CM_Container_MenuItems_WebChatAvailable_3miRM0000004CuZ_5823055</customLabel>

            <labelKey>CM_Container_MenuItems_WebChatAvailable</labelKey>
        </embeddedServiceCustomLabels>
        <itemName>Chat1</itemName>
        <osOptionsHideInIOS>false</osOptionsHideInIOS>
        <osOptionsHideInLinuxOS>true</osOptionsHideInLinuxOS>
        <osOptionsHideInMacOS>false</osOptionsHideInMacOS>
        <osOptionsHideInOtherOS>false</osOptionsHideInOtherOS>
        <osOptionsHideInWindowsOS>true</osOptionsHideInWindowsOS>
        <shouldOpenUrlInSameTab>false</shouldOpenUrlInSameTab>
    </embeddedServiceMenuItems>
    <embeddedServiceMenuItems>
        <channelType>Phone</channelType>
        <displayOrder>2</displayOrder>
        <itemName>Phone1</itemName>
        <osOptionsHideInIOS>true</osOptionsHideInIOS>
        <osOptionsHideInLinuxOS>false</osOptionsHideInLinuxOS>
        <osOptionsHideInMacOS>true</osOptionsHideInMacOS>
        <osOptionsHideInOtherOS>false</osOptionsHideInOtherOS>
        <osOptionsHideInWindowsOS>false</osOptionsHideInWindowsOS>
        <phoneNumber>1234567890</phoneNumber>
        <shouldOpenUrlInSameTab>false</shouldOpenUrlInSameTab>
    </embeddedServiceMenuItems>
    <embeddedServiceMenuItems>
        <channelType>CustomURL</channelType>
        <customUrl>https://google.com</customUrl>
        <displayOrder>3</displayOrder>
        <itemName>url1</itemName>
        <osOptionsHideInIOS>false</osOptionsHideInIOS>
        <osOptionsHideInLinuxOS>false</osOptionsHideInLinuxOS>
        <osOptionsHideInMacOS>false</osOptionsHideInMacOS>
        <osOptionsHideInOtherOS>false</osOptionsHideInOtherOS>
        <osOptionsHideInWindowsOS>false</osOptionsHideInWindowsOS>
        <shouldOpenUrlInSameTab>false</shouldOpenUrlInSameTab>
    </embeddedServiceMenuItems>
    <isEnabled>true</isEnabled>
    <masterLabel>ChannelMenuSettings</masterLabel>
    <site>SnapInCommunity</site>
</EmbeddedServiceMenuSettings>
```

Wildcard Support in the Manifest File

This metadata type doesn't support the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

EnablementMeasureDefinition

Represents an Enablement measure, which specifies the job-related activity that a user performs to complete a milestone or outcome in an Enablement program. A measure identifies a source object and optional related objects, with optional field filters and filter logic, for tracking the activity. To avoid deployment errors, deploy measures before you deploy programs.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

EnablementMeasureDefinition components have the suffix `.enablementMeasureDefinition` and are stored in the `enablementMeasureDefinitions` folder.

Version

EnablementMeasureDefinition components are available in API version 61.0 and later.

Special Access Rules

To access Enablement measures, the Design and Deliver Enablement Programs permission is required. This permission is available with the Enablement add-on license.

Fields

Field Name	Description
`description`	**Field Type** string **Description** An internal description for the measure to help Enablement admins understand the activity that's tracked.
`developerName`	**Field Type** string **Description** Required. The unique programmatic name for the measure record.
`masterLabel`	**Field Type** string

Field Name	Description
	Description Required. A user-friendly name for the measure, which is defined when the measure is created.
`sourceMeasureObject`	**Field Type** EnablementMeasureSourceObjectDefinition **Description** Required. The source object that tracks the activity you're measuring.
`status`	**Field Type** EnblProgramMeasureStatus (enumeration of type string) **Description** Required. Indicates whether the measure is published for use in Enablement programs. Values are: • `Draft`—The measure is saved, but not activated for use in programs. • `Published`—The measure is activated for use in programs. In Lightning Experience, this value is Active.

EnablementMeasureSourceObjectDefinition

Defines the source object, fields, field values, and calculation method for the job-related activity you're measuring.

Field Name	Description
`aggregateFieldApiName`	**Field Type** string **Description** The unique programmatic name for the field that the `aggregateFunction` uses for calculating. For example, if you're measuring how much revenue a sales rep has won, the value of `aggregateFunction` is `Sum` and the value of `aggregateFieldApiName` is `Amount`, which is the programmatic name of the Amount field on the Opportunity object.
`aggregateFunction`	**Field Type** EnablementAggregationType (enumeration of type string) **Description** Required. The method for calculating progress towards the milestone or outcome from records that qualify for the measure's criteria. Values are:

Field Name	Description
	• `Average` • `Count` • `Sum` For example, if you're measuring the number of deals won, the function is `Count`. If the function is `Average` or `Sum`, `aggregateFieldApiName` is required.
`dateFieldApiName`	**Field Type** string **Description** Required. The unique programmatic name for the field that defines when users get credit for the activity you're measuring. For example, if you're measuring the number of deals won, this value can be `ClosedDate`, the programmatic name of the Close Date field on the Opportunity object.
`displayFieldApiName`	**Field Type** string **Description** Required. The unique programmatic name for the field that primarily identifies records that qualify for the activity you're measuring. For example, if you're measuring the number of deals won, you're tracking the Opportunity object, and maybe you want to identify opportunities by their name. In this case, this field can be `Name`, the programmatic name of the Opportunity Name field on the Opportunity object.
`filterLogic`	**Field Type** string **Description** An expression that determines how to evaluate the optional field filters for the object.
`filters`	**Field Type** EnablementMeasureFilterDefinition[] **Description** The fields on the object and corresponding field values that further specify criteria for the activity you're measuring.
`objectApiName`	**Field Type** string **Description** Required. The unique programmatic name for the source object that tracks the activity you're measuring. For example, if you're measuring the number of deals won, this value is `Opportunity`, the programmatic name of the Opportunity object.

Field Name	Description
`relatedMeasureObjects`	**Field Type** EnablementMeasureRelatedObjectDefinition[] **Description** The optional related objects that further specify criteria for the activity you're measuring. Related objects can also specify additional filters.
`userFieldApiName`	**Field Type** string **Description** Required. The unique programmatic name for the field that defines who gets credit for the activity you're measuring. For example, if you're measuring the number of deals won by a sales rep, this value can be `OwnerId`, the developer name of the Opportunity Owner field on the Opportunity object.

EnablementMeasureFilterDefinition

Represents the fields on the source object or related objects and the corresponding field values that further specify criteria for the activity you're measuring.

Field Name	Description
`fieldApiName`	**Field Type** string **Description** Required. The unique programmatic name for the field that you're filtering by. For example, if you're tracking activity on the Opportunity object and want to filter by the Stage field, this value can be `StageName`.
`fieldValue`	**Field Type** string **Description** Required. The field value to filter by. For example, if you're tracking activity on the Opportunity object and want to filter by the Stage field, this value can be `Closed Won`.
`operator`	**Field Type** EnablementFilterOperator (enumeration of type string) **Description** Required. The logic for evaluating the specified field and field value. Values are: • `Contains`

Field Name	Description
	• DoesNotContain • DoesNotEqual • EndsWith • Equals • GreaterThan • GreaterThanOrEqual • In • IsNull • LessThan • LessThanOrEqual • NotIn • StartsWith
sequenceNumber	**Field Type** int **Description** Required. A number that specifies the order of the filter, relative to other filters, starting at 1.

EnablementMeasureRelatedObjectDefinition

Represents objects related to the source object. Related objects can further specify criteria for the activity you're measuring. Related objects can also have additional filters. For example, maybe you're measuring deals won for a specific product line. In this case, the source object is Opportunity, the related object is Opportunity Product, and the related object can have a filter for the specific product name.

Field Name	Description
filterLogic	**Field Type** string **Description** An expression that determines how to evaluate the optional field filters for the object.
filters	**Field Type** EnablementMeasureFilterDefinition[] **Description** The fields on the related object and the corresponding field values that further specify criteria for the activity you're measuring.
idFieldApiName	**Field Type** string

Field Name	Description
	Description Required. The programmatic name of the field that links the related object to the primary object. For example, if the primary object is Opportunity and the related object is Opportunity Product, this value is `OpportunityId`, the developer name of the Opportunity field on the Opportunity Product object.
`objectApiName`	**Field Type** string **Description** Required. The unique programmatic name for the related object. For example, if the related object is Opportunity Product, this value is `OpportunityLineItem`.

Declarative Metadata Sample Definition

The following is an example of an EnablementMeasureDefinition component.

```
<?xml version="1.0" encoding="UTF-8"?>
<EnablementMeasureDefinition xmlns="http://soap.sforce.com/2006/04/metadata">
    <description>Total amount in pipeline measure</description>
    <developerName>TotalAmountInPipeline</developerName>
    <masterLabel>Total Amount in Pipeline</masterLabel>
    <status>Draft</status>
    <sourceMeasureObject>
        <aggregateFieldApiName>Amount</aggregateFieldApiName>
        <aggregateFunction>Sum</aggregateFunction>
        <dateFieldApiName>CreatedDate</dateFieldApiName>
        <displayFieldApiName>Name</displayFieldApiName>
        <objectApiName>Opportunity</objectApiName>
        <userFieldApiName>OwnerId</userFieldApiName>
        <filters>
            <fieldApiName>StageName</fieldApiName>
            <fieldValue>Closed Won</fieldValue>
            <operator>Equals</operator>
            <sequenceNumber>1</sequenceNumber>
        </filters>
        <relatedMeasureObjects>
            <objectApiName>OpportunityLineItem</objectApiName>
            <idFieldApiName>OpportunityId</idFieldApiName>
            <filterLogic>1 OR 2</filterLogic>
            <filters>
                <fieldApiName>UnitPrice</fieldApiName>
                <fieldValue>10000</fieldValue>
                <operator>GreaterThan</operator>
                <sequenceNumber>1</sequenceNumber>
            </filters>
            <filters>
                <fieldApiName>TotalPrice</fieldApiName>
                <fieldValue>10000</fieldValue>
```

```
                <operator>GreaterThan</operator>
                <sequenceNumber>2</sequenceNumber>
            </filters>
        </relatedMeasureObjects>
    </sourceMeasureObject>
</EnablementMeasureDefinition>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>TotalAmountInPipeline</members>
        <name>EnablementMeasureDefinition</name>
    </types>
    <version>61.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

EnablementProgramDefinition

Represents an Enablement program, which includes exercises and measurable milestones to help users such as sales reps achieve specific outcomes related to your company's revenue goals.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

EnablementProgramDefinition components have the suffix `.enablementProgramDefinition` and are stored in the `enablementProgramDefinitions` folder.

Version

EnablementProgramDefinition components are available in API version 61.0 and later.

Special Access Rules

To access Enablement programs, the Design and Deliver Enablement Programs permission is required. This permission is available with the Enablement add-on license.

For partner programs in supported Experience Cloud sites, a supported Partner Relationship Management (PRM) add-on license is also required.

Fields

Field Name	Description
`description`	**Field Type** string **Description** Required. A summary of the program's goals and content that's visible to users.
`developerName`	**Field Type** string **Description** Required. The unique programmatic name for the program record.
`doesAllowSelfEnrollment`	**Field Type** boolean **Description** Indicates whether users can self-enroll in programs that are shared with them (`true`) or take only assigned programs (`false`). The default value is `false`.
`masterLabel`	**Field Type** string **Description** Required. A user-friendly name for the program, which is defined when the program is created.
`name`	**Field Type** string **Description** Required. The name of the program that's visible to users.
`network`	**Field Type** string **Description** The Experience Cloud site where a program is published for partner users.
`sections`	**Field Type** EnablementProgramSection[] **Description** Groups of milestones and exercises within a program.

Field Name	Description
`tasks`	**Field Type** EnablementProgramTask[] **Description** The outcome, milestones, and exercises in the program.
`type`	**Field Type** string **Description** Required. Indicates whether the program is for sales users in Lightning Experience (`Enablement`) or partner users in supported Experience Cloud sites (`PtnrEnablement`).

EnablementProgramSection

Represents a logical, trackable group of milestones and exercises within an Enablement program. When users take programs, they can expand or collapse sections.

Field Name	Description
`developerName`	**Field Type** string **Description** Required. The unique programmatic name for the section.
`name`	**Field Type** string **Description** Required. The title of the section that's visible to users when they take the program.
`sequenceNumber`	**Field Type** int **Description** Required. A number that specifies the order of the section, relative to other sections, starting at 0.
`tasks`	**Field Type** EnablementProgramTask[] **Description** The milestones and exercises in the section.

EnablementProgramTask

Represents an outcome, milestone, or exercise in an Enablement program. A program task is also known as a program item.

Field Name	Description
`day`	**Field Type** int **Description** Required. The day of the program when the item is due, relative to the program's start date. For example, if a user is expected to complete an exercise where they watch a product demo by day 2, this field's value is 2. For an outcome, this field specifies the number of days the full program takes. For example, if your program lasts 60 days, the value of this field is 60 for the outcome. This field's value contributes to the program's due date that users see when they take the program.
`description`	**Field Type** string **Description** Required. A summary of the outcome, milestone, or exercise that's visible to users when they take the program.
`developerName`	**Field Type** string **Description** Required. The unique programmatic name for the outcome, milestone, or exercise.
`exercise`	**Field Type** EnablementProgramTaskExercise **Description** The content used with an exercise. If `taskSubCategory` is `ActionItem`, this field isn't included when retrieving metadata.
`milestone`	**Field Type** EnablementProgramTaskMilestone **Description** The definition of an outcome or milestone, including the Enablement measures used and the criteria for completing the goal.
`name`	**Field Type** string

Field Name	Description
	Description Required. The title of the outcome, milestone, or exercise that's visible to users when they take the program.
`sequenceNumber`	**Field Type** int **Description** Required. A number that specifies the order of the milestone or exercise, relative to other milestones or exercises that have the same due date in the program or in the same section, starting at 0. This number determines the order of items that users see for that day in the program.
`taskCategory`	**Field Type** ProgramTaskDefCategory (enumeration of type string) **Description** Required. The type of the program item. Values are: • `Exercise` • `Milestone` `Milestone` is used for both the program's outcome and incremental milestones.
`taskSubCategory`	**Field Type** string **Description** Required. The type of exercise. This value determines the content associated with the exercise. For example, if the field value is `Video`, the exercise must reference video content from the Enablement workspace in the Digital Experiences app. Possible values are: • `ActionItem` • `AudioRecording` • `CustomExercise`—Available in API version 62.0 and later. • `Document` • `FeedbackRequest` • `Other` • `OtherExercise` • `ScheduledEvent` • `TextLesson` • `Trailhead` • `Video`

Field Name	Description
	When `taskCategory` is `Milestone`, the value of `taskSubCategory` must be `Other`.

EnablementProgramTaskExercise

Represents the content used with an exercise in an Enablement program.

Field Name	Description
`cmsContent`	**Field Type** EnablementProgramTaskCmsContent **Description** The definition of content managed in the Enablement workspace in the Digital Experiences app when `taskSubCategory` on EnablementProgramTask is `AudioRecording`, `Document`, `OtherExercise`, `ScheduledEvent`, `TextLesson`, or `Video`.
`customContent`	**Field Type** EnablementProgramTaskCustomContent **Description** The definition of content used with a custom exercise type when `taskSubCategory` on EnablementProgramTask is `CustomExercise`.
`externalContent`	**Field Type** EnablementProgramTaskExternalContent **Description** The definition of Trailhead content when `taskSubCategory` on EnablementProgramTask is `Trailhead`.
`feedbackContent`	**Field Type** EnablementProgramTaskFeedbackContent **Description** The definition of an assessment survey or Einstein prompt template when `taskSubCategory` on EnablementProgramTask is `FeedbackRequest`.

EnablementProgramTaskCmsContent

Defines content managed in the Enablement workspace in the Digital Experiences app for the Audio Recording, Document, Other, Scheduled Event, Text Lesson, or Video exercise types.

Field Name	Description
`apiName`	**Field Type** string **Description** Required in API version 62.0. The unique programmatic ID of the Digital Experiences content for the exercise. This string's format is ***`workspaceType`*`/`*`workspaceApiName.contentFQN`*`/`*`contentApiName`*, which matches the `fullName` field value on the corresponding DigitalExperience metadata type. For example, a Link content record from the Enablement workspace has this API name: `enablement/sfdcEnablement_EnablementWorkspace.sfdc_enablement__link/link_API_name`.
`contentKey`	**Field Type** string **Description** Not used in API version 62.0.

EnablementProgramTaskCustomContent

Defines content used with a custom exercise type.

Field Name	Description
`content`	**Field Type** string **Description** A serialized string returned by the Apex class that's specified in the corresponding LearningItemType metadata type's `apexSerializerDeserializer` field. This string identifies the content used with the custom exercise type so the custom exercise can be recreated in the destination org. This string: • Can't exceed 250 characters • Must contain only alphanumeric characters For details, see Implement Custom Exercise Types for Enablement Programs in the *Sales Programs and Partner Tracks with Enablement Developer Guide*.

EnablementProgramTaskExternalContent

Defines Trailhead content for the Trailhead exercise type.

Field Name	Description
`externalId`	**Field Type** string **Description** Required. The API name of the Trailhead module used with the exercise.
`providerType`	**Field Type** ProgramExtContentDefProvider (enumeration of type string) **Description** Required. The supported external content platform or system. Values are: • `Trailhead`

EnablementProgramTaskFeedbackContent

Defines the assessment survey or Einstein prompt template for the Feedback Request exercise type.

Field Name	Description
`inviteeCount`	**Field Type** int **Description** The number of peers or managers that the user is required to invite for giving feedback when `type` is `PeerFeedback`. Each peer or manager receives an invitation to the assessment survey associated with the Feedback Request exercise. When `type` is `AIFeedback`, this value is always `1`.
`promptTemplate`	**Field Type** string **Description** The prompt template to use with this exercise when `type` is `AIFeedback`.
`surveyDeveloperName`	**Field Type** string **Description** The unique programmatic name for the assessment survey that's sent to peers and managers when `type` is `PeerFeedback`.
`type`	**Field Type** string

Field Name	Description
	Description Required. The type of feedback used with the exercise. Values are: • `AIFeedback`—Users submit a video call, and Einstein generates feedback from the call's transcription. With this type, `promptTemplate` is required. • `PeerFeedback`—Users submit a URL to a sample of their work, and select peers and managers to review their work. Selected peers and managers complete an assessment survey. With this type, `surveyId` is required.

EnablementProgramTaskMilestone

Defines the requirements for an outcome or milestone, including the Enablement measures used for tracking activity and the criteria for completing the outcome or milestone.

Field Name	Description
`compositeMilestoneType`	**Field Type** EnblCompositeMilestoneType (enumeration of type string) **Description** The type of logic to use for evaluating the activity from two Enablement measures in a composite milestone. Values are: • `Addition` • `Division` • `Percentage`
`isMilestoneAnOutcome`	**Field Type** boolean **Description** Required. Indicates whether the program item is the program's outcome (`true`) or an incremental milestone (`false`).
`milestoneMeasures`	**Field Type** EnablementProgramTaskMilestoneMeasure[] **Description** The Enablement measures used with the outcome or milestone.
`milestoneTarget`	**Field Type** double

Field Name	Description
	Description The target value for a user to achieve to get credit for completing the outcome or milestone. The unit depends on the specific measure used with the outcome or milestone. For example, if the measure is the dollar amount of all closed opportunities, then the field value is measured in dollars.
`minimumSampleSize`	**Field Type** int **Description** The number of records to evaluate when calculating progress for an outcome or milestone that uses an average-based measure. Use this field with `milestoneTarget`. For example, if you want users to achieve an average deal size of $50,000 after closing 4 deals, then this field's value is `4` and `milestoneTarget` is `50000`.

EnablementProgramTaskMilestoneMeasure

Defines the Enablement measure used with an outcome or milestone.

Field Name	Description
`measureDefinitionDeveloperName`	**Field Type** string **Description** The unique programmatic name of the Enablement measure used with the outcome or milestone.
`sequenceNumber`	**Field Type** int **Description** A number that specifies the order of the Enablement measure when multiple measures are used with one outcome or milestone, starting at 0. For example, in a composite milestone that uses the Percentage function, the measure that provides the numerator value is sequence 0 and the measure that provides the denominator value is sequence 1.

Declarative Metadata Sample Definition

The following is an example of an EnablementProgramDefinition component.

```
<?xml version="1.0" encoding="UTF-8"?>
<EnablementProgramDefinition xmlns="http://soap.sforce.com/2006/04/metadata">
    <description>Get started with sales at Cloud Kicks and close your first
```

```
deal!</description>
    <developerName>Get_Started_Close_First_Deal_Program</developerName>
    <doesAllowSelfEnrollment>false</doesAllowSelfEnrollment>
    <masterLabel>Welcome to Sales at Cloud Kicks</masterLabel>
    <name>Welcome to Sales at Cloud Kicks</name>
    <sections>
        <developerName>section_0</developerName>
        <name>Learn the Ropes in Your First Week</name>
        <sequenceNumber>0</sequenceNumber>
        <tasks>
            <day>1</day>
            <description>Learn the basics of sales at Cloud Kicks.</description>
            <developerName>task_0</developerName>
            <exercise>
                <externalContent>
                    <externalId>sales-rep-training</externalId>
                    <providerType>Trailhead</providerType>
                </externalContent>
            </exercise>
            <name>Sales Rep Training</name>
            <sequenceNumber>0</sequenceNumber>
            <taskCategory>Exercise</taskCategory>
            <taskSubCategory>Trailhead</taskSubCategory>
        </tasks>
        <tasks>
            <day>2</day>
            <description>Watch our CEO explain the company vision.</description>
            <developerName>task_1</developerName>
            <exercise>
                <cmsContent>
                    <contentKey>MCFHN7EKKKUJBOPLXMJWZ7XWK3LA</contentKey>
                </cmsContent>
            </exercise>
            <name>See Our Company Vision</name>
            <sequenceNumber>1</sequenceNumber>
            <taskCategory>Exercise</taskCategory>
            <taskSubCategory>Video</taskSubCategory>
        </tasks>
        <tasks>
            <day>3</day>
            <description>Action Item</description>
            <developerName>task_2</developerName>
            <name>Action Item</name>
            <sequenceNumber>2</sequenceNumber>
            <taskCategory>Exercise</taskCategory>
            <taskSubCategory>ActionItem</taskSubCategory>
        </tasks>
        <tasks>
            <day>4</day>
            <description>Try out your first sales patch at Cloud Kicks and get feedback
from our in-house experts.</description>
            <developerName>task_3</developerName>
            <exercise>
                <feedbackContent>
```

```
                <inviteeCount>1</inviteeCount>
                <surveyDeveloperName>discovery_call_assessment</surveyDeveloperName>
            </feedbackContent>
            </exercise>
            <name>Feedback from Peers and Managers</name>
            <sequenceNumber>3</sequenceNumber>
            <taskCategory>Exercise</taskCategory>
            <taskSubCategory>FeedbackRequest</taskSubCategory>
        </tasks>
        <tasks>
            <day>5</day>
            <description>Complete a discovery calls by day 5.</description>
            <developerName>task_4</developerName>
            <isMilestoneAnOutcome>false</isMilestoneAnOutcome>
            <milestone>
                <milestoneMeasures>

<measureDefinitionDeveloperName>salesforceTemplate_CallsEmails</measureDefinitionDeveloperName>

                </milestoneMeasures>
                <milestoneTarget>1.0</milestoneTarget>
            </milestone>
            <name>Log a Discovery Call by Day 5</name>
            <sequenceNumber>4</sequenceNumber>
            <taskCategory>Milestone</taskCategory>
            <taskSubCategory>Other</taskSubCategory>
        </tasks>
        <tasks>
            <day>6</day>
            <description>Browse our sales leaders blog for more insights.</description>
            <developerName>task_5</developerName>
            <exercise>
                <cmsContent>
                    <contentKey>MC7VUB2BYTA5FKFLNNX7EYORYEYE</contentKey>
                </cmsContent>
            </exercise>
            <name>Review Tips from Sales Leaders</name>
            <sequenceNumber>5</sequenceNumber>
            <taskCategory>Exercise</taskCategory>
            <taskSubCategory>OtherExercise</taskSubCategory>
        </tasks>
        <tasks>
            <day>7</day>
          <description>Follow a screen flow for onboarding to the sales team.</description>

            <developerName>task_6</developerName>
            <exercise>
                <customContent>
                    <content>flowDeveloperName=OnboardingFlow</content>
                </customContent>
            </exercise>
            <name>Onboarding Flow</name>
            <sequenceNumber>6</sequenceNumber>
            <taskCategory>Exercise</taskCategory>
```

```
            <taskSubCategory>CustomExercise</taskSubCategory>
            <customSubCategoryName>ScreenFlowTaskSubCategory</customSubCategoryName>
        </tasks>
    </sections>
    <tasks>
        <day>30</day>
        <description>Close your first opportunity. Take make sure it's counted, set the
opportunity Stage field to Closed Won.</description>
        <developerName>task_enablementProgramOutcomeCard</developerName>
        <isMilestoneAnOutcome>true</isMilestoneAnOutcome>
        <milestone>
            <milestoneMeasures>

<measureDefinitionDeveloperName>measure_CloseFirstDeal</measureDefinitionDeveloperName>
            </milestoneMeasures>
            <milestoneTarget>1.0</milestoneTarget>
        </milestone>
        <name>outcome</name>
        <sequenceNumber>0</sequenceNumber>
        <taskCategory>Milestone</taskCategory>
        <taskSubCategory>Other</taskSubCategory>
    </tasks>
    <type>Enablement</type>
</EnablementProgramDefinition>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>Get_Started_Close_First_Deal_Program</members>
        <name>EnablementProgramDefinition</name>
    </types>
    <version>61.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

EnblProgramTaskSubCategory

Represents a custom exercise type that an Enablement admin adds to an Enablement program in Program Builder. A custom exercise type also requires a corresponding EnblProgramTaskDefinition record for Program Builder and corresponding LearningItem and LearningItemType records for when users take the exercise in the Guidance Center.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

EnblProgramTaskSubCategory components have the suffix `.enblProgramTaskSubCategory` and are stored in the `enblProgramTaskSubCategories` folder.

Version

EnblProgramTaskSubCategory components are available in API version 62.0 and later.

Special Access Rules

- For Enablement admins to create, update, and delete Enablement programs, the Design and Deliver Enablement Programs permission is required. This permission is enabled by default as part of the Manage Enablement Essentials permission set, which comes with the Enablement add-on license.
- For users who take Enablement programs, the Take Enablement Programs permission is required. This permission is enabled by default as part of the Use Enablement Programs permission set, which comes with the Enablement add-on license.

Important: Custom exercises aren't compatible with Partner Enablement programs.

Fields

Field Name	Description
`developerName`	**Field Type** string **Description** Required. The unique programmatic name for the EnblProgramTaskSubCategory record.
`icon`	**Field Type** string **Description** Required. The icon to use for the custom exercise type in Program Builder. Use the format ***`iconType:iconName`***, where the values correspond to icon categories and names from the Salesforce Lightning Design System. • ***`iconType`*** is the type of icon, such as `standard` or `doctype`. • ***`iconName`*** is the icon name, such as `flow` or `slide`. For example, to use the Standard type Flow icon, this value is `standard:flow`. For details, see Implement Custom Exercise Types for Enablement Programs in the *Sales Programs and Partner Tracks with Enablement Developer Guide*.

Field Name	Description
`learningItemType`	**Field Type** string **Description** Required. The programmatic name of the LearningItemType record that represents this custom exercise type in the Guidance Center when users take a program.
`masterLabel`	**Field Type** string **Description** Required. A user-friendly name for the EnblProgramTaskSubCategory, which is defined when it's created.

Declarative Metadata Sample Definition

The following is an example of an EnblProgramTaskSubCategory component for a custom exercise type that shows a screen flow.

```
<?xml version="1.0" encoding="UTF-8"?>
<EnblProgramTaskSubCategory xmlns="http://soap.sforce.com/2006/04/metadata">
    <developerName>ScreenFlowTaskSubCategory</developerName>
    <icon>standard:flow</icon>
    <learningItemType>ScreenFlowLearningItemType</learningItemType>
    <masterLabel>Screen Flow Exercise</masterLabel>
</EnblProgramTaskSubCategory>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>ScreenFlowTaskSubCategory</members>
        <name>EnblProgramTaskSubCategory</name>
    </types>
    <version>62.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

EntitlementProcess

Represents the settings for an entitlement process.

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

Entitlement process values are stored in files in the `entitlementProcesses` directory. Each file has the name of a process and the suffix `.entitlementProcess`. Each file contains one entitlement process or, if entitlement versioning is enabled, one version of an entitlement process.

The name of the file is the name of the entitlement process with the version appended to the end, if applicable (for example, an entitlement process named "gold_support" can have the file name "gold_support_v2.entitlementProcess"). This file name corresponds to the `slaProcess.NameNorm` field exposed through SOAP API. This file name is distinct from the `name` field, which represents what displays in the user interface and, if versioning is enabled, can be shared among multiple versions of the same entitlement process. The `slaProcess.NameNorm` field contains the lowercase version of the `name` field shown in the user interface.

Version

Entitlement processes are available in API version 27.0 and later.

Fields

Field Name	Field Type	Description
`active`	boolean	Indicates whether the entitlement process is active (`true`) or not (`false`).
`businessHours`	string	The business hours that apply to the entitlement process. This field is available in API version 30.0 and later.
`description`	string	The description of the entitlement process.
`entryStartDateField`	string	For milestone processes on which a case enters the process based on a custom date/time field on the case, specifies which date and time are used. Valid values are: • SlaStartDate (entitlement process start date) • CreatedDate (date case was opened) • ClosedDate (date case was closed) • LastModifiedDate (date case was last modified) • StopStartDate (date case was stopped)
`exitCriteriaBooleanFilter`	string	For milestone processes on which a case exits the process when custom criteria are met, and for which filter logic is added, specifies that logic.
`exitCriteriaFilterItems`	FilterItem[]	For milestone processes on which a case exits the process when custom criteria are met, specifies those criteria.
`exitCriteriaFormula`	string	For milestone processes on which a case exits the process when a custom formula evaluates to true, specifies that formula.

Field Name	Field Type	Description
isVersionDefault	boolean	Indicates whether the entitlement process is the default version (true) or not (false). This field is available in API version 28.0 and later.
milestones	EntitlementProcessMilestoneItem[]	Represents a milestone on the entitlement process.
name	string	The name of the entitlement process as it displays in the user interface.
SObjectType	string	Indicates the type of record that the entitlement process can run on.
versionMaster	string	Identifies the sequence of versions to which this entitlement process belongs. This field's contents can be any value as long as it's identical among all versions of the entitlement process. This field is available in API version 28.0 and later.
versionNotes	string	The description of the entitlement process version. This field is available in API version 28.0 and later.
versionNumber	int	The version number of the entitlement process. Must be 1 or greater. This field is available in API version 28.0 and later.

EntitlementProcessMilestoneItem

Represents a milestone item on an entitlement process.

Fields

Field Name	Field Type	Description
businessHours	string	The business hours that apply to the milestone. This field is available in API version 30.0 and later.
criteriaBooleanFilter	string	For milestones that apply only when criteria are met and for which filter logic is added, specifies that logic.
milestoneCriteriaFilterItems	FilterItem[]	For milestones that apply only when criteria are met, specifies those criteria.
milestoneCriteriaFormula	string	For milestones that apply only when a formula evaluates to true, specifies that formula.
milestoneName	string	The name of the milestone.

Field Name	Field Type	Description
minutesCustomClass	string	The name of the Apex class that is used to calculate the trigger time. This field is available in API version 30.0 and later.
minutesToComplete	int	The number of minutes from when the case enters the entitlement process that the milestone occurs.
successActions	WorkflowActionReference[]	The actions triggered when the milestone is completed.
timeTriggers	EntitlementProcessMilestoneTimeTrigger[]	The time triggers on an entitlement process milestone.
useCriteriaStartTime	boolean	When the milestone starts: when the milestone criteria are met (true) or when the case enters the entitlement process (false).

EntitlementProcessMilestoneTimeTrigger

Represents the time trigger on an entitlement process milestone.

Fields

Field Name	Field Type	Description
actions	WorkflowActionReference[]	The actions to take when the time trigger is reached, if, at that time, the milestone isn't completed.
timeLength	int	The length of time between the time trigger activation and the milestone target completion date. This length of time can be a negative or positive value. Negative values indicate that the target completion date hasn't yet arrived and correspond to warning time triggers. Positive values indicate that the target completion date has passed and correspond to violation time triggers.
workflowTimeTriggerUnit	MilestoneTimeUnits (enumeration of type string)	Specifies the type of unit used to determine when a workflow is triggered. Valid values are: • Minutes • Hours • Days

Declarative Metadata Sample Definition

Here's a sample entitlement process.

```
<?xml version="1.0" encoding="UTF-8"?>
<EntitlementProcess xmlns="http://soap.sforce.com/2006/04/metadata">
    <active>true</active>
    <description>eppersone</description>
```

```
    <entryStartDateField>SlaStartDate</entryStartDateField>
    <exitCriteriaBooleanFilter>1 OR 2</exitCriteriaBooleanFilter>
    <exitCriteriaFilterItems>
        <field>Case.IsClosed</field>
        <operation>equals</operation>
        <value>true</value>
    </exitCriteriaFilterItems>
    <exitCriteriaFilterItems>
        <field>Case.Description</field>
        <operation>startsWith</operation>
        <value>foo</value>
    </exitCriteriaFilterItems>
    <milestones>
        <milestoneName>m1</milestoneName>
        <minutesToComplete>1</minutesToComplete>
        <successActions>
            <name>emailBob</name>
            <type>Alert</type>
        </successActions>
        <timeTriggers>
            <actions>
                <name>emailAlice</name>
                <type>Alert</type>
            </actions>
            <actions>
                <name>setEscalateToTrue</name>
                <type>FieldUpdate</type>
            </actions>
            <timeLength>1</timeLength>
            <workflowTimeTriggerUnit>Minutes</workflowTimeTriggerUnit>
        </timeTriggers>
        <timeTriggers>
            <actions>
                <name>setStopToTrue</name>
                <type>FieldUpdate</type>
            </actions>
            <timeLength>2</timeLength>
            <workflowTimeTriggerUnit>Minutes</workflowTimeTriggerUnit>
        </timeTriggers>
        <useCriteriaStartTime>false</useCriteriaStartTime>
    </milestones>
    <milestones>
        <milestoneCriteriaFilterItems>
            <field>Case.Priority</field>
            <operation>equals</operation>
            <value>High</value>
        </milestoneCriteriaFilterItems>
        <milestoneName>m2</milestoneName>
        <minutesToComplete>120</minutesToComplete>
        <useCriteriaStartTime>true</useCriteriaStartTime>
        <successActions>
            <name>emailBob</name>
            <type>Alert</type>
        </successActions>
```

```
    </milestones>
</EntitlementProcess>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

EntitlementTemplate

Represents an entitlement template. Entitlement templates are predefined terms of customer support that you can quickly add to products. For example, you can create entitlement templates for Web or phone support so that users can easily add entitlements to products offered to customers.

EntitlementTemplate extends the Metadata metadata type and inherits its `fullName` field.

Declarative Metadata File Suffix and Directory Location

EntitlementTemplate components are stored in the `entitlementTemplates` directory of the corresponding package directory. The file name matches the unique name of the entitlement template, and the extension is `.entitlementTemplate`.

Version

Lightning Platform EntitlementTemplate components are available in API version 18.0 and higher.

Fields

Field	Field Type	Description
`businessHours`	string	The entitlement's supported business hours.
`casesPerEntitlement`	int	The total number of cases the entitlement supports.
`entitlementProcess`	string	The entitlement process associated with the entitlement. Entitlement processes are timelines that include all the steps (milestones) that your support team must complete to resolve cases. Each process includes logic to determine how to enforce the correct service level for your customers.
`isPerIncident`	boolean	`true` if entitlements created from this template service a limited number of cases; `false` otherwise.
`term`	int	The number of days the entitlement is in effect.
`type`	string	The type of entitlement, such as Web or phone support.

Declarative Metadata Sample Definition

A sample XML definition of an entitlement template is shown below.

```
<?xml version="1.0" encoding="UTF-8"?>
<EntitlementTemplate xmlns="http://soap.sforce.com/2006/04/metadata">
    <businessHours>AlternateBusinessHours</businessHours>
    <casesPerEntitlement>12</casesPerEntitlement>
    <entitlementProcess>Process1</entitlementProcess>
    <isPerIncident>true</isPerIncident>
    <term>33</term>
    <type>Phone Support</type>
</EntitlementTemplate>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

EscalationRules

Represents case escalation rules to escalate cases automatically if they aren't resolved within a certain time. You can access rules metadata for all applicable objects, for a specific object, or for a specific rule on a specific object.

The `package.xml` syntax for accessing all escalation rules for all objects is:

```
    <types>
        <members>*</members>
        <name>EscalationRules</name>
    </types>
```

All rules for a specific object use a similar syntax without the wildcard. For example, all escalation rules for the Case object would use this syntax:

```
    <types>
        <members>Case</members>
        <name>EscalationRules</name>
    </types>
```

You can also access specific escalation rules for an object. The following example only accesses the "samplerule" and "newrule" escalation rules on the Case object. Notice that for this example the type name syntax is `EscalationRule` and not `EscalationRules`.

```
    <types>
        <members>Case.samplerule</members>
        <members>Case.newrule</members>
        <name>EscalationRule</name>
    </types>
```

File Suffix and Directory Location

EscalationRules for an object have the suffix `.escalationRules` and are stored in the `escalationRules` folder. For example, all Case escalation rules are stored in the `Case.escalationRules` file.

Version

EscalationRules components are available in API version 27.0 and later.

Fields

Field Name	Field Type	Description
`escalationRule`	EscalationRule[] on page 963	Represents one escalation rule and specifies whether it's active or not. Escalation rules are processed in the order they appear in the EscalationRules container.

EscalationRule

Field Name	Field Type	Description
`active`	boolean	Indicates whether the escalation rule is active (`true`) or not (`false`).
`fullname`	string	Inherited from Metadata, this field is defined in the WSDL for this metadata type. It must be specified when creating, updating, or deleting. See `createMetadata()` to see an example of this field specified for a call. This value can't be `null`.
`ruleEntry`	`RuleEntry[]`	Contains the definitions of the rule entries in the escalation rule.

RuleEntry

Represents the fields used by the rule.

Field Name	Field Type	Description
`booleanFilter`	string	Advanced filter conditions that were specified for the rule.
`businessHours`	string	The hours when escalation actions are performed. Specify only if `businessHoursSource` is set to `Static`.
`businessHoursSource`	BusinessHoursSourceType (enumerations of type string)	Valid values are: • `None` • `Case` • `Static`
`criteriaItems`	FilterItem	The items in the list that define the assignment criteria.
`disableEscalationWhenModified`	boolean	Indicates whether the escalation is disabled when the record is modified `true`) or not (`false`).

Field Name	Field Type	Description
escalationAction	EscalationAction[]	The actions to perform when the escalation criteria are met.
escalationStartTime	EscalationStartTimeType (enumeration of type string)	Indicates the start time for the escalation. Valid values are: • CaseCreation • CaseLastModified
formula	string	The validation formula. Specify either formula or criteriaItems, but not both fields.

EscalationAction

Describes the action to take for an escalation rule.

Field Name	Field Type	Description
assignedTo	string	The name of the user or queue the item is assigned to.
assignedToTemplate	string	Specifies the template to use for the email that is automatically sent to the new owner specified by the escalation rule. Lightning email templates aren't packageable. We recommend using a Classic email template.
assignedToType	AssignToLookupValueType (enumeration of type string)	Valid values are: • User • Queue
minutesToEscalation	int	The number of minutes until the escalation occurs.
notifyCaseOwner	boolean	Indicates that the owner of the case is notified when the case is escalated true) or not (false).
notifyEmail	string	Specifies the email address of the user to notify.
notifyTo	string	Specifies the user to notify.
notifyToTemplate	string	Specifies the template to user for the notification email.

Declarative Metadata Sample Definition

The following is an example EscalationRules component:

```
<EscalationRules xmlns="http://soap.sforce.com/2006/04/metadata">
    <escalationRule>
        <fullName>samplerule</fullName>
        <active>false</active>
```

```
        <ruleEntry>
            <businessHours>test</businessHours>
            <businessHoursSource>Static</businessHoursSource>
            <criteriaItems>
                <field>Case.Description</field>
                <operation>contains</operation>
                <value>test</value>
            </criteriaItems>
            <escalationAction>
                <assignedTo>someuser@org.com</assignedTo>
                <assignedToTemplate>emailtemplatename</assignedToTemplate>
                <assignedToType>User</assignedToType>
                <minutesToEscalation>1440</minutesToEscalation>
                <notifyCaseOwner>false</notifyCaseOwner>
            </escalationAction>
            <escalationStartTime>CaseLastModified</escalationStartTime>
        </ruleEntry>
    </escalationRule>
</EscalationRules>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

EventDelivery

Represents how an event instance maps to a target payload. Removed in API version 46.0. This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

Event delivery components have the suffix file path `.delivery`, and are stored in the `eventDeliveries` folder.

Version

Event delivery components are available in API versions 41.0 to 45.0.

Limits

Your org can have a maximum of 2500 EventDelivery object instances.

Fields

Field Name	Field Type	Description
`eventParameters`	EventParameterMap[]	An array of parameters to deliver in addition to the published event's data.

Field Name	Field Type	Description
eventSubscription	string	Required. The ID of the subscription to deliver the data to.
referenceData	string	User-defined non-unique identifier.
type	EventDeliveryType (enumeration of type string)	Required. Determines what action occurs when the event is delivered to the listeners on behalf of the subscribers. Valid values are: • StartFlow—When the event occurs, it's delivered to a flow of type CustomEvent. Those flows are built through Process Builder. • ResumeFlow—Reserved for future use.

EventParameterMap

Parameters to deliver in addition to the published event's data.

If type is StartFlow, you must include a parameter where parameterName is FlowVersionName and parameterValue is the name of the flow that you want to start. The flow name must include its version number. For example, myFlow-3.

Each event delivery can have up to 10 parameters.

Field Name	Field Type	Description
parameterName	string	The parameter name.
parameterValue	string	The parameter value.

Declarative Metadata Sample Definition

The following is an example of an event delivery file.

```
<?xml version="1.0" encoding="UTF-8"?>
<EventDelivery xmlns="http://soap.sforce.com/2006/04/metadata">
   <eventParameters>
      <parameterName>FlowVersionName</parameterName>
      <parameterValue>My_Event_Based_Process-1</parameterValue>
   </eventParameters>
   <eventSubscription>MySubscription</eventSubscription>
   <referenceData>My_Event_Based_Process_1</referenceData>
   <type>StartFlow</type>
</EventDelivery>
```

The following is an example package.xml that deploys or retrieves all the available event delivery metadata in your org.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
   <types>
      <members>EventDelivery</members>
      <name>*</name>
   </types>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

EventRelayConfig

Represents the configuration of an event relay, which relays platform events and change data capture events from Salesforce to Amazon EventBridge.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

`EventRelayConfig` components have the suffix `.eventRelay` and are stored in the `eventRelays` folder.

Version

EventRelayConfig components are available in API version 56.0 and later.

Special Access Rules

- You must have the Customize Application permission to deploy and retrieve this type.
- You can update only the `state` and `relayOption` fields and not `eventChannel` or `destinationResourceName`.

Fields

Field Name	Description
`destinationResourceName`	**Field Type** string **Description** Required. The developer name of the named credential, which stores the AWS account information. The `destinationResourceName` value contains the `callout:` prefix. For example: `callout:MyRelayNamedCredential`
`eventChannel`	**Field Type** string **Description** Required. The full name of the event channel used in the event relay. For example: `MyRelayChannel__chn`

Field Name	Description
`label`	**Field Type** string **Description** The label for the event relay. The label is displayed in the user interface. Make sure you use a meaningful label that describes your event relay and try to make it unique.
`relayOption`	**Field Type** string **Description** A JSON-encoded string that contains an option for resuming an event relay after the system recovers from an error. This option is used if the event relay can't resume after the last relayed event. The options available are: • `{\"ReplayRecovery\":\"LATEST\"}`—(Default) Start relaying events from new events received in the event bus. Use this option if you aren't interested in missed events while the relay was down. • `{\"ReplayRecovery\":\"EARLIEST\"}`—Resend all events stored in the event bus and relay new events thereafter. The event bus stores events for up to three days. Use this option if you want to reprocess all stored events and catch up on missed events.
`state`	**Field Type** EventRelayAdminState (enumeration of type string) **Description** The execution state of the event relay. Possible values are: • `RUN`—The event relay is running and actively relaying event messages from Salesforce to Amazon EventBridge. • `PAUSE`—An administrator paused the event relay. No events are relayed to Amazon EventBridge during this status. All current state information is saved. • `STOP`—(Default) The event relay is stopped and no events are relayed to Amazon EventBridge. All current state information is deleted. The event relay is created with a default state of `STOP` if you don't specify this field. If you specify this field when creating an event relay, the only valid value you can set is `STOP`. • `DELETE`—Reserved for future use.
`usageType`	**Field Type** string **Description** Reserved for future use.

Declarative Metadata Sample Definition

The following is an example of an EventRelayConfig component with the file name `Carbon_Comparison_Relay.eventRelay`.

```
<?xml version="1.0" encoding="UTF-8"?>
<EventRelayConfig xmlns="http://soap.sforce.com/2006/04/metadata">
    <destinationResourceName>callout:AWS_Account</destinationResourceName>
    <eventChannel>Carbon_Comparison_Channel__chn</eventChannel>
    <label>Carbon Comparison Relay</label>
    <relayOption>{\"ReplayRecovery\":\"LATEST\"}</relayOption>
    <state>STOP</state>
</EventRelayConfig>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>Carbon_Comparison_Relay</members>
        <name>EventRelayConfig</name>
    </types>
    <version>64.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

EventSubscription

Represents a subscription to an event type. Removed in API version 46.0. This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

EventSubscription components have the suffix file path `.subscription`, and are stored in the `eventSubscriptions` folder.

Version

Event subscription components are available in API versions 41.0 to 45.0.

Limits

Your org can have a maximum of:

- 4,000 total event subscriptions
- 2,000 active event subscriptions

Fields

Field Name	Field Type	Description
`active`	boolean	If the subscription isn't active, it never receives any events.
`eventParameters`	EventParameterMap[]	An array of parameters that must be true for published events.
`eventType`	string	Required. The name of the platform event.
`referenceData`	string	Required. If the subscriber is a flow of type CustomEvent, `referenceData` is ***`flowName_versionNumber`***. For example, `Printer_Management_2`.

EventParameterMap

An array of parameters that must be true for published events. For example, subscribe to Vendor Response events only if `Status__c` is `Shipped`.

Each event subscription can have up to 10 parameters.

Field Name	Field Type	Description
`parameterName`	string	Required. The published event's field name.
`parameterValue`	string	The value that must be true.

Declarative Metadata Sample Definition

The following is an example of an active event subscription.

```
<?xml version="1.0" encoding="UTF-8"?>
<EventSubscription xmlns="http://soap.sforce.com/2006/04/metadata">
    <active>true</active>
    <eventType>Printer_Status__e</eventType>
    <referenceData>Printer_Management</referenceData>
</EventSubscription>
```

The following is an example of an inactive event subscription that sets event parameters.

```
<?xml version="1.0" encoding="UTF-8"?>
<EventSubscription xmlns="http://soap.sforce.com/2006/04/metadata">
   <name>MySubscription</name>
   <active>false</active>
   <eventParameters>
      <parameterName>Ink_Status__c</parameterName>
      <parameterValue>low</parameterValue>
   </eventParameters>
   <eventParameters>
      <parameterName>Serial_Number__c</parameterName>
      <parameterValue>00123456789</parameterValue>
   </eventParameters>
   <eventType>Printer_Status__e</eventType>
```

```
    <referenceData>My_Event_Based_Process_1</referenceData>
</EventSubscription>
```

The following is an example `package.xml` that deploys or retrieves all the available event subscription metadata in your org.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>EventSubscription</name>
    </types>
    <version>41.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

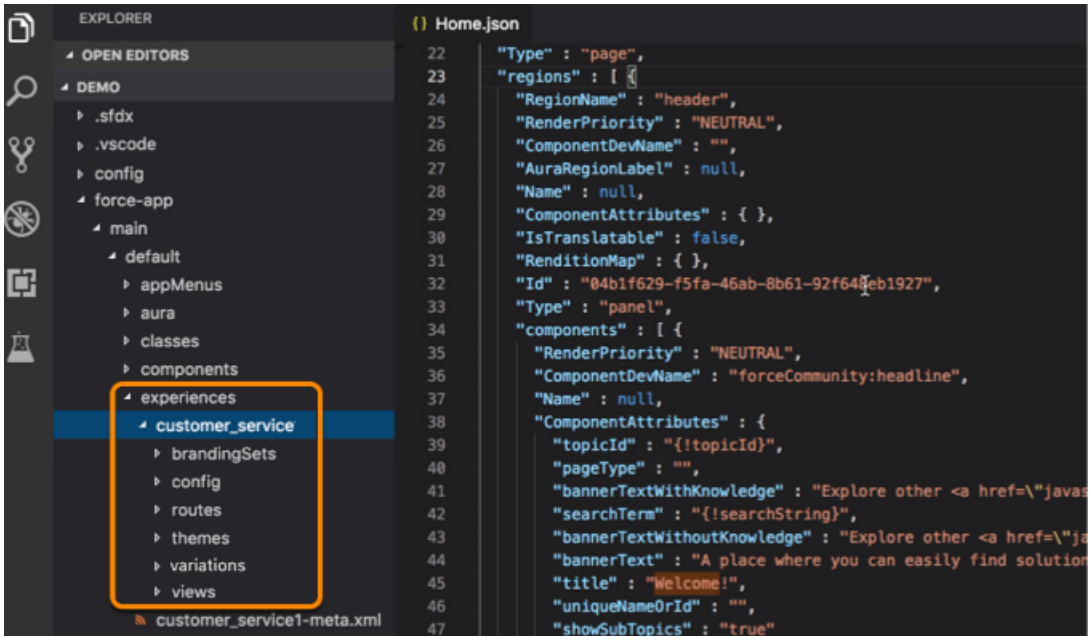
ExperienceBundle

Represents a text-based code structure of the settings and site components, such as pages, branding sets, and themes that make up an Experience Builder site. Developers can quickly update and deploy Experience Builder sites *programmatically* using their preferred development tools. This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

ExperienceBundle components have the suffix `.json` and are stored in the `experiences` folder when retrieved. Each Experience Builder site in your org has its own folder. Each of these folders contains other folders for the supported properties.

The ExperienceBundle can contain one or more site definitions under the `experiences` folder. Each site definition has resource folders for brandingSets, config, routes, themes, variations, and views, each with additional, related configuration information in JSON files. Here's an example site definition, showing the resource folders.



Version

ExperienceBundle components are available in API version 46.0 and later.

Special Access Rules

To use the ExperienceBundle metadata type for Aura-based Experience Builder sites, from Setup, enter *Digital Experiences* in the Quick Find box, and then select **Settings**. Select **Enable ExperienceBundle Metadata API**, and save your changes. LWR sites use ExperienceBundle by default.

Fields

Field Name	Field Type	Description
`experienceResources`	ExperienceResources[]	The list of resources in this ExperienceBundle. Each resource represents an artifact of a site such as brandingSets, config, routes, themes, variations, and views.
`label`	string	Required. Represents the name of the ExperienceBundle.
`type`	SiteType (enumeration of type string)	Required. Identifies the kind of site. Only Experience Builder sites are supported, using the value `ChatterNetworkPicasso`.
`urlPathPrefix`	string	Specify a URL prefix for an Experience Builder site. For example, in the site URL SitesSubdomainName.force.com/customers, customers is the UrlPathPrefix. Note: For authenticated LWR sites created before Winter '23 and Aura sites, the URL path prefix ends in /s, and the part of the path without the /s must match the Network metadata type's URL. For unauthenticated LWR sites and authenticated LWR sites created after

Field Name	Field Type	Description

		Winter '23 through Experience Builder or Connect API, this path doesn't contain /s, and the path can be anything as long as there's no conflict.
		Sample meta.xml file
		`<?xml version="1.0" encoding="UTF-8"?>`
		`<ExperienceBundle`
		`xmlns="http://soap.sforce.com/2006/04/metadata">`
		`<label>SampleStarterSite2</label>`
		`<type>ChatterNetworkPicasso</type>`
		`<urlPathPrefix>SampleStarterSite2/s</urlPathPrefix>`
		`</ExperienceBundle>`

ExperienceResources

Represents a list of sites in the bundle.

Field Name	Field Type	Description
`experienceResource`	ExperienceResource[]	The list of resources in this ExperienceBundle. Each resource represents a property for the site, such as brandingSets, config, routes, themes, and views.

ExperienceResource

Represents specific site information included in the ExperienceBundle.

Each type has a folder in the structure. Each folder contains one or more files providing information about that type and the site. Each corresponds to a specific folder and file in the ExperienceBundle.

Field Name	Field Type	Description
`fileName`	string	Required. Name of resource file.
`format`	string	Required. Only `JSON` is allowed.
`source`	base64	The `JSON` content of each file.
`type`	string	Required. The type of the resource. Valid values are: • `brandingSets` • `config` • `routes` • `themes` • `views`

Folders and Bundled Definitions

Each ExperienceBundle includes folders and associated data that is contained in JSON files.

brandingSets Folder

This folder contains one JSON file per branding set, named *brandingSets_name*.json. Each file has the same structure and properties.

<brandingSets_name>.json

Property	Type	Description
brandingSetType	string	Required in LWR sites. Not applicable for Aura sites. Represents whether the color palette stored in the branding set is for the entire site or a specific section. You can't change one branding set type to another. Available in API Version 52.0 and later. Valid values are: • APP: The branding set applies to the entire site. There can be only one branding set of this type. • SCOPED: The branding set applies to a specific section.
definitionName	string	Required. Represents the name for the branding set that is used in grouping branding sets under a theme. Defined as *theme*:branding-*theme*. For example, if the site theme is Stella, the definitionName would be stella:branding-stella. In addition, there are several standard templates that have unique naming: • Customer Account Portal uses cpt:branding-cpt • Customer Service uses service:branding-service • Help Center uses helpCenter:branding-helpCenter • Partner Central uses prm:branding-prm • Build Your Own uses starter:branding-starter Note: The combination of definitionName + label must be unique in your org.
id	UUID	Represents the component's GUID.
label	string	Represents the name of the branding set. Note: The combination of definitionName + label must be unique in your org.
type	string	Represents the component type. The only supported value is brandingSet.

Property	Type	Description
values	map	Required. Represents a map of branding values that can be applied to a site.

```
{
  "values" : {
    "HeaderBackgroundColor" : "#FFFFFF",
    "TextTransformStyle" : "none",
    "BorderColor" : "#D4D4D4",
    "DetailTextColor" : "#5A5A5A",
    "HeaderFonts" : "Ek Mukta",
    "CardBackgroundColor" : "rgba(255, 255, 255, 0)",
    "LoginBackgroundColor" : "#F4F4F4",
    "_ActionColorTrans" : "rgba(25, 124, 190, 0.9)",
    "LoginBackgroundImage" :
"../../../../sfsites/picasso/core/external/salesforceIdentity/images/background.jpg?v=1",

    "PageBackgroundColor" : "#F5F7FA",
    "_HeaderTextColor" : "rgba(34,34,34,.8)",
    "_NavigationMenuHoverColor" : "rgba(255,255,255,.2)",
    "_HeaderInputBackgroundColor" : "rgba(255,255,255,.4)",
    "TextColor" : "#222222",
    "NavigationMenuTextColor" : "#222222",
    "_HeaderPlaceholderTextColor" : "rgba(85,85,85,.8)",
    "_OverlayTextColorShadow" : "#000000",
    "ActionColor" : "#0099DE",
    "CompanyLogo" : "",
    "_LinkColorDarker" : "#135F90",
    "_ActionColorDarker" : "#135F90",
    "_HoverColor" : "rgba(25, 124, 190, 0.05)",
    "ErrorFontColor" : "#ff9e9e",
    "OverlayTextColor" : "#FFFFFF",
    "PrimaryFont" : "Ek Mukta",
    "LinkColor" : "#3558D6"
    },
  "definitionName" : "cpt:branding-cpt",
  "label" : "Customer Account Portal",
  "id" : "283407c3-5938-4a6b-b97f-621cda6968c8",
  "type" : "brandingSet"
 }
```

config Folder

The `config` folder contains several JSON files.

- *sitename*.json
- languages.json
- nativeConfig.json
- *page_name*.json

Note: One for each single-page application in the site: `loginAppPage.json` and `mainAppPage.json`

sitename`.json` **File Properties**

Property	Type	Description
`authenticationType`	string	For LWR sites, indicates whether guest users have access to the site. Note: For Aura sites, use `isAvailableToGuests` instead. Valid values are: • `AUTHENTICATED`: The site isn't public. Only authenticated users can access the site after logging in. • `AUTHENTICATED_WITH_PUBLIC_ACCESS_ENABLED`: The site is an authenticated site, but the **Public can access the site** checkbox is enabled in Experience Builder in **Settings** > **General**. Guest users can access the site. • `UNAUTHENTICATED`: The unauthenticated site is publicly available to anyone on the web, and doesn't support login or authentication. Guest users can access the site. `UNAUTHENTICATED` isn't supported for LWR sites created after Winter '23 through Experience Builder or Connect API. To allow guest user access, we recommend using `AUTHENTICATED_WITH_PUBLIC_ACCESS_ENABLED`. Available in API version 51.0 and later.
`forgotPasswordRouteId`	UUID	Represents the ID of the route to use when a user forgets their password. Note: Unsupported if the active Experience Builder template for the site doesn't support login (such as Help Center).
`isAvailableToGuests`	boolean	For Aura sites, indicates whether public users have access to the site (`true`) or not (`false`). The default value is `false`. Note: For LWR sites, use `authenticationType` instead.
`isFilteredComponentsView`	boolean	Indicates whether the list of components is filtered based on the current page type (`true`) or not (`false`). Some components require specific parameters from the page and don't work unless you manually configure them. The default value is `false`.
`isLockerServiceEnabled`	boolean	Indicates whether Lightning Locker is enabled (`true`) or disabled (`false`). The default value is `true`. Available in API version 55.0 and later.
`isProgressiveRenderingEnabled`	boolean	Indicates whether the display order of page components is prioritized (`true`) or not (`false`). The default value is `false`.
`loginAppPageId`	UUID	Represents the ID of the login page. Note: Unsupported if the active Experience Builder template for the site doesn't support login (such as Help Center).
`mainAppPageId`	UUID	Required. Represents the ID of the main page.

Property	Type	Description
`preferredDomain`	string	Represents the name of the domain to use for indexing a site's pages. Improves search engine results. Available in API version 48.0 and later.
`preferredDomainId`	string	Represents the domain to use for indexing a site's pages. Improves search engine results. Removed in API version 48.0. Use `preferredDomain` instead.
`selfRegistrationRouteId`	UUID	Represents the ID of the login route to use for self-registration. Note: Unsupported if the active Experience Builder template for the site doesn't support login (such as Help Center).
`type`	string	Represents the component type. The only supported value is `site`.

trustedSitesForScript container

When implemented, there's one `trustedSitesForScript` container in *`sitename`*`.json`.

Property	Type	Description
`id`	UUID	Represents the component's GUID.
`isActive`	boolean	Indicates if allowlisted item is active (`true`) and must be respected or inactive (`false`) and must not be treated as an allowlisted source. Default is `false`.
`trustedSiteName`	string	Name of the allowlisted source as it appears in the UI.
`trustedSiteUrl`	string	The fully qualified URL of the allowlisted source.
`type`	string	Represents the component type. The only supported value is `trustedSitesForScripts`.

```
{
   "isAvailableToGuests" : false,
   "isFilteredComponentsView" : false,
   "mainAppPageId" : "df9907cb-6e68-4ca1-8bb2-51173ca5374e",
   "loginAppPageId" : "58e9939a-84b2-498d-bbc5-7a89d89087fa",
   "selfRegistrationRouteId" : "ad5c8bf1-297f-4ad3-b47c-0e35d85f10ef",
   "forgotPasswordRouteId" : "e3139f6f-44d8-4eec-be9d-3609ce063039",
   "isProgressiveRenderingEnabled" : false,
   "preferredDomain" : "none",
   "selfRegistrationRouteId" : "b8fe8ab1-f266-41e1-a63b-4791165f3c1d",
   "trustedSitesForScript" : [ {
     "id" : "92c489e2-0b7b-4a48-9c88-bef7e8fe6f1b",
     "isActive" : true,
     "trustedSiteName" : "test",
     "trustedSiteUrl" : "https://123.com",
     "type" : "trustedSitesForScripts"
```

```
    }, {
      "id" : "92c489e2-0b7b-4a48-9c88-bef7e8fe6f1c",
      "isActive" : true,
      "trustedSiteName" : "test1",
      "trustedSiteUrl" : "https://1234.com",
      "type" : "trustedSitesForScripts"
  } ],
    "type" : "site"
}
```

`languages.json` **File Properties**

Property	Type	Description
`defaultCode`	string	Required. Represents the base language code plus the country code where used.
`defaultLabel`	string	Required. Defines the display label for the language.
`id`	UUID	Represents the component's GUID.
`type`	string	Represents the component type. The only supported value is `languageContainer`.

There's one section per supported language as a container in `languages.json`

language container

Property	Type	Description
`countryCode`	string	Represents the country code of the selected language. This string can be empty. It applies only when the selected language has variations depending on the country, like Arabic (Algeria) and Arabic (Bahrain). In this case, use `countryCode` to distinguish between them. For example: `{ languageCode" : "ar", "CountryCode" : "DZ", "Label" : "Arabic (Algeria) (DZ)",}, { "Code" : "ar", "CountryCode" : "BH", "Label" : "Arabic (Bahrain) (BH)",}`
`fallbackLanguageId`	UUID	Represents the language to use when no content is available for the selected language. For example, if a site visitor chooses **Japanese** from the language selector, but there's no content for that page in Japanese, then content is displayed in the fallback language. Only one level of fallback is allowed for LWR sites. Here are examples for an LWR site where English is the default language, and Spanish, French, and Finnish are available site languages. • Not allowed: Spanish falls back to French, and French falls back to Finnish. This configuration includes two levels of fallback. • Allowed: Spanish falls back to French, and French falls back to English. This configuration is allowed because English is the site's default language.

Property	Type	Description
		• Allowed: Spanish falls back to French, and French has no fallback. This configuration includes only one level of fallback.
`id`	UUID	Represents the component's GUID.
`isActive`	boolean	Indicates whether a language is available to site visitors in the language selector (`true`) or not (`false`). The default value is `true`.
`label`	string	Defines the display label for a language. The display label appears in any language selector components that you add to your site and in the language selector in Experience Builder.
`languageCode`	string	Represents the language code for the selected language.
`type`	string	Represents the component type. The only supported value is `language`.

```
{
    "defaultCode" : "en_US",
    "defaultLabel" : "English (US)",
    "id" : "04597c83-0b9d-4f16-9f4d-4ec28bd553b4",
    "type" : "languageContainer",
    "languages" : [ {
        "languageCode" : "af",
        "countryCode" : "",
        "isActive" : true,
        "label" : "Afrikaans",
        "fallbackLanguageId" : "c6e7fe67-55e0-47b3-ad58-bf49539249f0",
        "id" : "22036d6f-11ce-4f7b-b7f0-f2c409f817ea",
        "type" : "language"
        }
    ]
}
```

The page file represents single-page applications in the site. One file per page, named *`page_name`*`.json`.

Note: Each Experience Builder site is actually a single-page application, which is a web app that loads a single `HTML` page. Single-page applications use multiple views to update the page dynamically as the user interacts with it.

`nativeConfig.json` **File Properties**

Property	Type	Description
`showHamburgerMenu`	boolean	Required. Controls whether the hamburger menu is shown.
`mobilePublisherAppUpdateConfig`	boolean	Required. Controls whether and which App Version Update message is shown. To avoid service disruptions, users must be on the app version that supports enhanced domains.
`id`	UUID	Represents the component's GUID.

Property	Type	Description
type	string	Represents the component type. The only supported value is nativeConfig.

```
{
 "id": "a70a0e5e-0400-4531-94dc-8f587daa5946",
 "nativeMobileNavConfig": {
   "showBackButton": true,
   "showHamburgerMenuWithBackButton": false
 },

 "mobilePublisherAppUpdateConfig": {
   "enableAppUpdate" : true,
   "forceAppUpdate" : true,
   "minVersion" : {
       "ios" : {
           "version" : "10.0"
 },
       "android" : {
               "version" : "10.1"
 }
 }
 },
 "nativeTabMenu": {
   "branding": {
     "iconTintColorUnselected": "#C9C5C5",
     "barTintColor": "#FF00FF",
     "iconTintColor": "#555321"
   },
   "menuItems": [
     {
       "iconAsset": "icon_homepng",
       "targetUrl": "/"
     },
     {
       "name": "Test",
       "iconAsset": "icon_filespng",
       "targetUrl": "/files"
     }
   ]
 },
 "showNavMenu": true,
 "type": "nativeConfig"
}
```

nativeMobileNavConfig container

A required container for the configuration for the Native Navigation Bar component.

Property	Type	Description
showBackButton	boolean	Controls whether the Back button is shown on iOS devices.

Property	Type	Description
`showHamburgerMenuWithBackButton`	boolean	Controls whether the hamburger menu is shown, in addition to the Back button, on iOS devices.

mobilePublisherAppUpdateConfig container

A required container for the configuration of the App Version Update message.

Property	Type	Description
`enableAppUpdate`	boolean	Controls whether the App Version Update message is shown, to encourage users to update by giving them a choice of whether to do so. Set the properties to `"enableAppUpdate" : true`, and `"forceAppUpdate" : false` to show the message that encourages your users to update. If you don't want to show an update message, for example if all your users are on the correct version or your site uses a custom domain, set the property to `"enableAppUpdate" : false`, and don't use the `forceAppUpdate` property.
`forceAppUpdate`	boolean	Controls whether the App Version Update message to require users to update is shown. Set the properties to `"enableAppUpdate" : true`, and `"forceAppUpdate" : true` to show the message that requires your users to update.
`minVersion`	string	Controls the iOS and Android Minimum App Versions. These property values are currently hard coded to ensure that the app versions supporting enhanced domains are used.

nativeTabMenu container

A required container for the configuration of the hamburger menu and Back button behavior.

Property	Type	Description
`branding`	map	Settings for the Native Navigation Bar component branding. Valid keys are: • iconTintColorUnselected • iconTintColor • barTintColor Supply a valid 6 digit hexadecimal as the value for all properties.
`menuItems`	list	Items which must be displayed in the Native Navigation Bar component.

menuItems container

A container within the nativeTabMenu container that specifies the items displayed in the tab bar of the Native Navigation Bar component.

Property	Type	Description
`name`	string	Optional. The label of the tab bar menu item.
`targetUrl`	string	Required. The relative URL to which the tab bar menu item points.
`iconAsset`	string	Required. Name of the ContentAsset to use for the tab bar menu item.

page_name`.json` File Properties

Property	Type	Description
`cmsSettings`	map	Settings for the CMS Connect header and footer. Valid values are: • `headerName` • `headerUrl` • `headerPersonalization` • `footerName` • `footerUrl` • `footerPersonalization` Both source and target org must have the CMSConnect and CMSPersonalization org perms enabled for settings to be retrieved.
`currentThemeId`	UUID	Required. Represents the UUID of the site's current theme. This field is available for `mainAppPage.json` and `loginAppPage.json` (where applicable).
`headMarkup`	string	Required. Allows the addition of custom markup to the site's main page `<head>` tag. Similar to using **Experience Builder** > **Setting** > **Advanced** > **Head Markup** See Salesforce Help for markup guidance.
`id`	UUID	Required. Represents the component's GUID.
`isRelaxedCSPLevel`	boolean	Controls the ability to run scripts and script access to third-party hosts. The default is `false`. This field is available for `mainAppPage.json` and `loginAppPage.json` (where applicable).
`label`	string	Required. Represents the name of the page.
`templateName`	string	Required. The unique developer name of the template. Allowed values include: • CPT Community Template (which represents the Customer Account Portal template) • Help Center Template (which represents the Help Center template) • microsite-template-marketing (which represents the Microsite (LWR) template • PRM Community Template (which represents the Partner Central template)

Property	Type	Description
		• Service Community Template (which represents the Customer Service template) • Starter Template (which represents the Build Your Own (Aura) template) • talon-template-byo (which represents the Build Your Own (LWR) template) • *Custom_template_name* (which is the name of a customized template that was exported as a Bolt Solution) Alternatively, you can retrieve a list of allowed template name values using Connect REST API. See Experience Builder Templates in the *Connect REST API Developer Guide*.
`type`	string	Required. Represents the component type. The only supported value is `appPage`.

```
{
    "headMarkup" : null,
    "isRelaxedCSPLevel" : false,
    "templateName" : "Starter Template",
    "cmsSettings" : { },
    "currentThemeId" : "ff52089c-6ad9-4dd9-b5b5-251d4a117ce3",
    "label" : "main",
    "id" : "df9907cb-6e68-4ca1-8bb2-51173ca5374e",
    "type" : "appPage"
}
```

routes Folder

The `routes` folder contains one JSON file per page, named *<page_name>*`.json`.

<page_name>`.json`

Property	Type	Description
`activeViewId`	UUID	Required. Represents the default view of the route. Used when there are no defined audiences or the user doesn't match any audience. Available in API version 48.0 and later.
`appPageId`	UUID	Required. Represents the single page application (SPA) page for the route. It points to either `main.json` or `login.json`.
`configurationTags`	string[]	Required. Represents the configuration tags for the route. The only supported value is `allow-in-static-site`. Available in API Version 51.0 and later. Note: This is an internal property and must not be edited.
`devName`	string[]	Required. Represents the unique API name that's defined when creating a new route. Available in API version 59.0 and later.

Property	Type	Description
`id`	UUID	Required. Represents the component GUID. Inherited from the component.
`label`	string	Required. Represents the name of the route. Inherited from the component.
`objectApiName`	string	Required. The name of the custom object API. (Not available for standard objects.)
`pageAccess`	string	Required. Identifies the status of a route as public or private. When set to the default value `UseParent`, the status of the site determines the status of the route. Not editable from the user interface for routes that are always private. Valid values are `UseParent`, `Public`, and `RequiresLogin`.
`routeType`	string	Required. Identifies the type of route. Value is unique among all routes that share the same SPA page. The value in `viewType` must match.
`type`	string	Required. Represents the component type. The only supported value is `route`.
`urlPrefix`	string	Required. Represents the base URL for the route.

```
{
    "urlPrefix" : "",
    "appPageId" : "b5fe94e2-071f-47b2-b76d-427a624cb407",
    "configurationTags" : "allow-in-static-site"
    "routeType" : "home",
    "pageAccess" : "UseParent",
    "label" : "Home",
    "id" : "c7263124-7bc4-4147-a39a-25fe7e305b98",
    "type" : "route"
}
```

themes Folder

The `themes` folder contains one JSON file per theme named *`theme_name`*`.json`.

`theme_name``.json`

Property	Type	Description
`activeBrandingSetId`	UUID	The id of the branding set currently in use. The branding set's `definitionName` must match the theme's `brandingSetReference`.
`customCSS`	string	Custom CSS for pages created in the Experience Builder template.
`developerName`	string	Required. The unique developer name of the theme. Most themes derive their names directly, for example Jepson uses `jespon` for its `developerName`. Standard templates have unique values: • `cpt` for Customer Account Portal • `service` for Customer Service • `helpCenter` for Help Center

Property	Type	Description
		• `prm` for Partner Central • `starter` for Build Your Own
`id`	UUID	Required. Represents the component's GUID.
`label`	string	Represents the name of the theme.
`layouts`	map	Required. Maps `ThemeLayoutType` to UUID, and contains the definition of the ThemeLayout. Login and Inner theme layouts are always required.
`type`	string	Required. Represents the component type. The only supported value is `theme`.

```
{
    "developerName" : "cpt",
    "layouts" : {
        "Login" : "12162c3e-06ac-43a9-adc7-db36ae5140b0",
        "Inner" : "c09d58be-0622-4fc4-806a-ed34174929f9"
    },
    "customCSS" : "",
    "activeBrandingSetId" : "283407c3-5938-4a6b-b97f-621cda6968c8",
    "label" : "Customer Account Portal",
    "id" : "ff52089c-6ad9-4dd9-b5b5-251d4a117ce3",
    "type" : "theme",
    "views" : [ {
        "componentName" : "salesforceIdentity:loginBody2",
        "label" : "Login",
        "id" : "12162c3e-06ac-43a9-adc7-db36ae5140b0",
        "type" : "view",
        "regions" : [ {
            "regionName" : "header",
            "id" : "f8354922-11f2-495d-9d89-0a51943af2b0",
            "type" : "region",
            "components" : [ ]
        } ]
    } ]
}
```

Note: Views can be children of a theme. These children are structured the same as views in the views folder.

variations Folder

Experience variations let you change the default behavior of the Experience Builder site based on the audience. The `variations` folder contains one JSON file per experience variation. The file is named *`experienceVariation_name`*`.json`.

Note:

- Experience variations are available in API version 47.0 and later.
- The name of your JSON file must match the `developerName` of your variation to avoid issues when deploying a site more than one time.

Four distinct types of variations are supported: branding sets, page variations, component visibility, and component attributes. The different variations are indicated through the `componentVariant` container.

For example, you want the site to show a page variation for the home page when a user meets certain audience criteria. To achieve this, create an audience and then target that audience to your experience variation using `targetId` in the `componentVariant` container of the experience variation definition file.

`experienceVariation_name``.json`

Property	Type	Description
`componentVariants`	list	Required. A list of component variants that belong to this experience variation. Note: Only one component variant per experience variation is allowed.
`developerName`	string	Required. The unique developer name of the experience variation. This name is used in the `targetValue` field of a Personalization API target and can't be updated after it's set. Note: For more information, see Audience.
`id`	UUID	Required. Represents the GUID of the component.
`type`	string	Required. Represents the type of the component. The only supported value is `experienceVariation`.

When implemented, there's one container in each *`experienceVariation_name`*`.json` file describing the variation.

componentVariant container

Property	Type	Description
`id`	UUID	Required. Represents the GUID of the component.

`propertyOverrides`	map	Required. Defines the property overrides for the given theme, route, or component `targetId`.
		For example, if the `targetId` is pointing to a theme, you can override the `defaultBrandingSet` property of the theme to use a different branding set for this experience variation.
		Supported property overrides:
		`activeBrandingSetId`
		Defines which branding set to use when `targetId` is a theme. Uses the format:
		`"activeBrandingSetId" : "ID_of_brandingset"`
		`activeViewId`
		Defines which page variation to use when `targetId` is a route. Uses the format:
		`"activeViewId" : "ID_of_view"`

Property	Type	Description

		componentAttributes
		Supported only for CMS Collection components and navigation components, such as Navigation Menu or Tile Menu. Components can be placed in header and footer regions, and also in the view body.
		• Defines which navigation linkset to display when `targetId` is a navigation component.
		The value of the property is a JSON container with a single key-value pair denoting the attribute and the value of the attribute.
		`NavigationMenuEditorRefresh` is the only supported attribute. Uses the format:
		`"componentAttributes" : { "NavigationMenuEditorRefresh" : "linkset_name" }`
		• Defines which content collection to display when `targetId` is a CMS Collection component.
		The value of the property is a JSON container with a single key-value pair denoting the path to the attribute and the value of the attribute. `config/dataProviderDefinition/attributes/dataProviderInfo/apiName` is the only supported attribute. Uses the format:
		`"componentAttributes" : { "config/dataProviderDefinition/attributes/dataProviderInfo/apiName":"collection_name" }`
		isVisible
		Defines whether a component is visible for the audience when `targetId` is a component. Unsupported for components in header or footer regions. Uses the format:
		`"isVisible": boolean`
		Note:
		• Only one entry in the map is allowed.
		• For a component, you can vary either its visibility or attributes but not both together.

`targetId`	UUID	Required. The UUID of the item whose properties you're overriding. Must be the ID of a theme, route, or component.
`type`	string	Required. Represents the type of the component. The only supported value is `experienceVariation`.

Example of an experience variation for a branding set

```
{
   "id": "64e93604-78fa-11e9-8f9e-2a86e4085a59",
   "developerName": "BrandingVariation",
   "type": "experienceVariation",
   "componentVariants": [{
      "id": "4bf0af78-8d73-11e9-bc42-526af7764f64",
      "type": "componentVariant",
      // Theme UUID
      "targetId": "c810858e-78fa-11e9-8f9e-2a86e4085a59",
      "propertyOverrides": {
         // Brandingset UUID
         "activeBrandingSetId": "be9f4760-78fa-11e9-8f9e-2a86e4085a59"
      }
   }]
}
```

Example of an experience variation for a page variation

```
{
   "id": "64e93604-78fa-11e9-8f9e-2a86e4085a59",
   "developerName": "PageVariation",
   "type": "experienceVariation",
   "componentVariants": [{
      "id": "4bf0af78-8d73-11e9-bc42-526af7764f64",
      "type": "componentVariant",
      // Route UUID
      "targetId": "c810858e-78fa-11e9-8f9e-2a86e4085a59",
      "propertyOverrides": {
         // View UUID
         "activeViewId": "be9f4760-78fa-11e9-8f9e-2a86e4085a59"
      }
   }]
}
```

Example of an experience variation for component visibility

```
{
   "id": "64e93604-78fa-11e9-8f9e-2a86e4085a59",
   "developerName": "ComponentVisibilityVariation",
   "type": "experienceVariation",
   "componentVariants": [{
      "id": "4bf0af78-8d73-11e9-bc42-526af7764f64",
      "type": "componentVariant",
      // Component UUID
      "targetId": "c810858e-78fa-11e9-8f9e-2a86e4085a59",
      "propertyOverrides": {
         "isVisible": true
      }
   }]
}
```

Example of a component variation for a CMS Collection component

```
{
   "id" : "6ce1260f-cb01-45a0-8947-f2d85602a3db"
   "developerName": "Home_CMS_Collection_Component_Properties",
   "type": "experienceVariation",
   "componentVariants": [{
      "id" : "3gh1260f-cb01-45a0-8947-f2d92037a4db"
      "type": "componentVariant",
      "targetId": "d77369e6-7230-43e7-9b59-6e91c47b3273",
      "propertyOverrides": {
         "componentAttributes": {

"config/dataProviderDefinition/attributes/dataProviderInfo/apiName":"SilverCollection"
         }
      },
   }],
}
```

Example of a component variation for Navigation Menu component

```
{
   "id" : "8cf943b8-525d-4c13-a719-6ebc7d61a81e",
   "developerName" : "Default_Navigation_Menu_Component_Properties",
   "type" : "experienceVariation",
   "componentVariants" : [{
      "id" : "5be1260f-cb01-45a0-8947-f2d85602a4db",
      "type" : "componentVariant",
      "targetId" : "fdf9eb51-ddc5-4e79-9ea8-5b94f5ca8db4",
      "propertyOverrides" : {
         "componentAttributes" : {
            "NavigationMenuEditorRefresh" : "NavMenu1"
         }
      },
   }],
}
```

views Folder

The `views` folder contains several JSON files that each define a view. Each Experience Builder site is built from single-page applications, which are web apps that load a single HTML page. Single-page applications consist of multiple views that update the page dynamically as the user interacts with it.

A *view* is made up of *regions* that contain other regions or *components* in the rendered page for the user. Within the `views` folder there's one file per view, named *`view_name`*`.json`.

Note: Single-page applications in your site are defined in the page files of the `config` folder.

`view_name``.json`

Property	Type	Description
`appPageId`	UUID	Required. Single page application (SPA) page ID of the view. It points to either `main.json` or `login.json`.

Property	Type	Description
`componentName`	string	Required. The FQN of the layout component. The component must implement `forceCommunity:layout` or, for theme layouts, `forceCommunity:themeLayout`
`id`	UUID	Required. Represents the GUID of the component.
`label`	string	Required. The name that appears in **Experience Builder** > **Settings** > **Theme** > **Configure**.
`themeLayoutType`	string	Theme layout type of the view (exposed only for views).
`type`	string	Required. Represents the type of the component. The only supported value is `view`.
`viewType`	string	Required. Matches `routeType` for the route.

There are one or more regions as a container in each *<view_name>*`.json`

region container

Property	Type	Description
`id`	UUID	Required. Represents the component GUID.
`regionLabel`	string	Specifies region labels for tabs. Note: This property is present only for tab regions that are children of a component.
`regionName`	string	Required. Matches the design attribute in the design file of the layout component.
`type`	string	Required. Represents the component type. The only supported value is `region`.

Each *<view_name>*`.json` file contains a hidden region called `sfdcHiddenRegion`. The hidden region contains a component that represents the SEO assistant component. In Aura sites, the component's definition is `forceCommunity:seoAssistant`, and in LWR sites, the component's definition is `community_builder:seoAssistant`. This component corresponds to the SEO page properties that you can configure in Experience Builder and isn't visible on your pages. To improve search engine results, use the SEO assistant component to set the `customHeadTags`, `description`, and `pageTitle` properties for your public and custom site pages. You can't edit the other properties associated with the SEO assistant component. To learn more about what the title, description, and custom head tags properties represent and which head tags are allowed, see SEO Page Properties in Experience Builder.

There are one or more components as a container in the region section of each *<view_name>*`.json`

component container

Property	Type	Description
`componentAttributes`	HashMap	Required. The design attribute values of the component.

Property	Type	Description
`componentName`	string	Required. The FQN of the component. Only components that can be used in the component panel in Experience Builder can be used in this field.
`id`	UUID	Required. Represents the component GUID. Note: If you add a component to ExperienceBundle, you can enter any value because the system automatically generates a UUID for the component when deployed.
`renderPriority`	enums.priority	Sets priority value for progressive rendering of the component. Possible Values: `HIGHEST`, `HIGH`, `NEUTRAL` Note: Only evaluated if the site has progressive rendering turned on in **Experience Builder** > **Settings** > **Advanced**.
`renditionMap`	HashMap	Map of different rendition keys to UUIDs of RenditionComponents.
`scopedBrandingSetID`	UUID	Required for LWR sites. Not applicable for Aura sites. Represents the ID of a branding set for a specific `community_layout:section` component. Available in API Version 52.0 and later.
`type`	string	Required. Represents the component type. The only supported value is `component`.

Each component can have a rendition container in each *<view_name>*`.json`

rendition container

Property	Type	Description
`id`	UUID	Required. Represents the component GUID.
`renditionValue`	map	Map of different variations of a component, such as different languages of text.
`type`	string	Required. Represents the component type. The only supported value is `renditionComponent`.

```
{
    "themeLayoutType" : "Inner",
    "viewType" : "account-management",
    "appPageId" : "df9907cb-6e68-4ca1-8bb2-51173ca5374e",
    "componentName" : "siteforce:sldsOneColLayout",
    "label" : "Account Management",
    "id" : "9ca8fa47-8e87-4915-a6f7-c2d8d37f3076",
    "type" : "view",
    "regions" : [ {
         "regionName" : "content",
         "id" : "969ada98-7d72-4e45-8a10-7db51fae247c",
         "type" : "region",
         "components" : [ {
```

```
            "componentName" : "forceCommunity:tabset",
            "componentAttributes" : {
                "tabsetConfig" :
"{\"UUID\":\"4711850e-ffdc-4375-a45e-f716bcdbbb1c\",\"activeTab\":\"tab1\",

\"useOverflowMenu\":false,\"tabs\":[{\"UUID\":\"bc8fb51f-4783-43d4-9376-60c07677a367\",\"tabName\":\"Members\",

\"tabKey\":\"tab1\",\"locked\":false,\"allowGuestUser\":false,\"seedComponents\":[{\"fqn\":\"forceCommunity:relatedList\",

\"attributes\":{\"parentRecordId\":\"{!CurrentUser.accountId}\",\"relatedListName\":\"Users\",\"customTitle\":\"Members\",

\"showCustomTitle\":\"true\",\"showBreadCrumbs\":\"false\",\"showRowNumbers\":\"false\",\"showManualRefreshButton\":\"false\"}}]},

{\"UUID\":\"f2793a99-b757-4be4-846f-dc98a13a8139\",\"tabName\":\"Branding\",\"tabKey\":\"tab2\",\"locked\":false,

\"allowGuestUser\":false,\"seedComponents\":[{\"fqn\":\"forceCommunity:accountBrandRecord\",

                \"attributes\":{\"recordId\":\"{!CurrentUser.accountId}\"}}]}]}",
                "regions" : ""
            },
            "renderPriority" : "NEUTRAL",
            "renditionMap" : { },
            "id" : "4711850e-ffdc-4375-a45e-f716bcdbbb1c",
            "type" : "component",
            "renditions" : [ {
               "renditionValue" : {
                  "LumenInstanceAttributes" : {
                  "richTextValue" : "<p>new text</p>"
                  }
               },
              "id" : "9d8878df-f520-4010-861c-57b930a3daab",
              "type" : "renditionComponent"
            } ]
        } ]
    } ]
}
```

Declarative Metadata Sample Definition

Here's an example of an ExperienceBundle declaration. For individual folder and file examples for the bundled code, see brandingSets, config, routes, themes, variations, and views.

```
<xsd:complexType name="ExperienceBundle">
    <xsd:complexContent>
        <xsd:extension base="tns:Metadata">
            <xsd:sequence>
                <xsd:element name="experienceResources" minOccurs="0"
type="tns:ExperienceResources"/>
```

```
                <xsd:element name="label" type="xsd:string"/>
                <xsd:element name="type" type="tns:SiteType"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
    <xsd:complexType name="ExperienceResources">
        <xsd:sequence>
            <xsd:element name="experienceResource" minOccurs="0" maxOccurs="unbounded"
type="tns:ExperienceResource"/>
        </xsd:sequence>
    </xsd:complexType>
<xsd:complexType name="ExperienceResource">
    <xsd:sequence>
        <xsd:element name="fileName" type="xsd:string"/>
        <xsd:element name="format" type="xsd:string"/>
        <xsd:element name="source" minOccurs="0" type="xsd:base64Binary"/>
        <xsd:element name="type" type="xsd:string"/>
    </xsd:sequence>
</xsd:complexType>
```

Usage

Tip: Before you update the .json files of an Experience Builder site, we recommend making a copy of the site's folder as a backup.

When you add a component to ExperienceBundle, you can enter any value for the `id`, because the system automatically generates a UUID for the component when deployed.

When deploying an Experience Builder site with ExperienceBundle, ensure that the SiteDotCom type isn't included in the manifest file.

ExperienceBundle doesn't support retrieving and deploying across different API versions. If you're trying to upgrade ExperienceBundle metadata from an earlier API version to a later one—for example, from API version 48.0 to 49.0—take the following steps:

1. Set the API version in the package.xml manifest file to 48.0 and deploy the package.
2. Then, set the API version in package.xml to 49.0.
3. To get the latest ExperienceBundle updates, retrieve the package.

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SEE ALSO:

ExperienceBundleSettings

Developer Guide: ExperienceBundle for Experience Builder Sites

ExperiencePropertyTypeBundle (Beta)

Represents a property type. When you create a custom property type for a Lightning web component, deploy this bundle to your org.

Note: This feature is a Beta Service. Customer may opt to try such Beta Service in its sole discretion. Any use of the Beta Service is subject to the applicable Beta Services Terms provided at Agreements and Terms.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Structure and Directory Location

ExperiencePropertyTypeBundle components are stored in the experiencePropertyTypeBundles folder. Here's an example of how the folder is structured.

```
+--myMetadataPackage
    +--experiencePropertyTypeBundles (1)
        +--addressProperty (2)
            +--schema.json (3)
            +--design.json (4)
```

- In the experiencePropertyTypeBundles folder (1) is a folder for each custom property type.
- Each custom property type folder is named in the format propertyTypeName. In this example (2), the name is addressProperty.
- Each propertyTypeName folder contains a JSON file or files that define the property type.
  - A `schema.json` file (3), which is a JSON schema that drives the property type validation
  - An optional `design.json` file (4), which provides the user experience and property editor information for that property type

Version

ExperiencePropertyTypeBundle components are available in API version 58.0 and later.

Special Access Rules

The ExperiencePropertyTypeBundle metadata type is available only for use with Lightning web components on LWR sites.

Fields

Field Name	Description
`description`	**Field Type** string **Description** Explanatory text about the property type.
`masterLabel`	**Field Type** string

Field Name	Description
	Description Required. A user-friendly name for ExperiencePropertyTypeBundle, which is defined when the ExperiencePropertyTypeBundle is created.
`resources`	**Field Type** ExperiencePropertyTypeBundleResource[] **Description** A list of source files in the experiencePropertyTypeBundles folder.

ExperiencePropertyTypeBundleResource

Represents a resource inside ExperiencePropertyTypeBundle.

Field Name	Description
`fileName`	**Field Type** string **Description** Required. The file name of the resource.
`filePath`	**Field Type** string **Description** Required. The file path of the resource.
`source`	**Field Type** base64Binary **Description** Required. The content of the resource.

Declarative Metadata Sample Definition

This `package.xml` file retrieves all the ExperiencePropertyTypeBundle components in an org.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>ExperiencePropertyTypeBundle</name>
    </types>
    <version>58.0</version>
</Package>
```

In the retrieved `.zip` file, each property type is nested under an experiencePropertyTypeBundles folder.

This example shows the directory structure in the `.zip` file of a property type named addressProperty.

```
experiencePropertyTypeBundles
    addressProperty
        schema.json
        design.json
```

Here are the contents of the files in the addressProperty directory. The addressProperty is a complex type that includes subproperties for firstName, lastName, address, city, state, and postal code. Each subproperty is a primitive type.

Contents of `schema.json`:

```
{
 "title": "Simple Address Type",
 "lightning:type": "lightning__objectType",
 "properties": {
   "firstName": {
     "lightning:type": "lightning__textType",
     "title": "First Name"
   },
   "lastName": {
     "lightning:type": "lightning__textType",
     "title": "Last Name"
   },
   "address": {
     "lightning:type": "lightning__textType",
     "title": "Address Line 1"
   },
   "city": {
     "lightning:type": "lightning__textType",
     "title": "City"
   },
   "state": {
     "lightning:type": "lightning__textType",
     "title": "State"
   },
   "postalCode": {
     "lightning:type": "lightning__numberType",
     "title": "Postal Code"
   }
 },
 "required": ["firstName", "lastName"]
}
```

Contents of `design.json` (an optional file):

```
{
 "definition": "lightning/tabsetLayout",
 "children": [
   {
     "definition": "lightning/tabLayout",
     "attributes": {
        "label": "First Tab"
     },
     "children": [
```

```
        {
          "definition": "lightning/propertyLayout",
          "attributes": {
            "property": "aProperty"
          }
        },
        {
          "definition": "lightning/propertyLayout",
          "attributes": {
            "property": "bProperty"
          }
        },
     ]
    },
      {
      "definition": "lightning/tabLayout",
      "attributes": {
         "label": "Second Tab"
      },
      "children": [
        {
          "definition": "lightning/propertyLayout",
          "attributes": {
            "property": "cProperty"
          }
        },
      ]
    },
  ]
}
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SEE ALSO:

External Link: Custom Property Types and Property Editors (Beta)

ExplainabilityMsgTemplate

Represents information about the template that contains the decision explanation message for a specified expression set step type.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

ExplainabilityMsgTemplate components have the suffix `.explainabilityMsgTemplate` and are stored in the `ExplainabilityMsgTemplates` folder.

Version

ExplainabilityMsgTemplate components are available in API version 56.0 and later.

Fields

Field Name	Description
`evaluationResult`	**Field Type** EvaluationResult (enumeration of type string) **Description** Required. The type of result for which the message template can be used. The step type for which the result is evaluated can be a condition, conditional group, or branch. Valid values are: • `Failed` • `Passed` • `NoResult`
`expressionSetStepType`	**Field Type** ExpressionSetStepType (enumeration of type string) **Description** Required. The step type in an expression set that uses the explainability message template. Valid values are: • `Aggregation` • `Branch` • `BusinessElement` • `Calculation` • `Condition` • `DecisionTableLookup` • `ListEnabledGroup` • `ListFilter` • `MatrixLookup` • `ReferenceProcedure`

Field Name	Description
`expsSetProcessType`	**Field Type** ExpsSetProcessType (enumeration of type string) **Description** Required. The type of industry that's using the expression set. Valid values are: • `Bre` • Note: When Business Rules Engine is enabled for a Salesforce instance, the default value is '`Bre`'. Other usage types may be available to you depending on your industry solution and permission sets.
`isDefault`	**Field Type** boolean **Description** Indicates whether the decision explainer template for a specified step type is default (true) or not (false).
`masterLabel`	**Field Type** string **Description** Required. Master label the for ExplainabilityMsgTemplate.
`message`	**Field Type** string **Description** Required. The message associated with the template for a specific expression set step type.

Declarative Metadata Sample Definition

The following is an example of an ExplainabilityMsgTemplate component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ExplainabilityMsgTemplate
 xmlns="http://soap.sforce.com/2006/04/metadata">
 <evaluationResult>Passed</evaluationResult>
 <expressionSetStepType>Condition</expressionSetStepType>
 <expsSetProcessType>ProductQualification</expsSetProcessType>
 <isDefault>false</isDefault>
 <masterLabel>ML EMT testDM</masterLabel>
```

```
 <message>EMT Testing</message>
</ExplainabilityMsgTemplate>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package
 xmlns="http://soap.sforce.com/2006/04/metadata">
 <types>
  <members>*</members>
  <name>ExplainabilityMsgTemplate</name>
 </types>
 <version>64.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ExpressionSetDefinition

Represents an expression set definition.

Note: Before deploying an expression set or an expression set version to a target org, review these Expression Set Migration Considerations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

ExpressionSetDefinition components have the suffix `.expressionSetDefinition` and are stored in the `expressionSetDefinition` folder.

Version

ExpressionSetDefinition components are available in API version 55.0 and later.

Fields

Field Name	Description
`description`	**Field Type** string

Field Name	Description
	Description The description of an expression set definition.
`executionScale`	**Field Type** ExpsSetExecutionScale (enumeration of type string) **Description** Specifies the scale of the inputs that an expression set processes. The scale determines where the expression set is executed. Valid values are: • `High` • `Low` Available in API version 61.0 and later.
`interfaceSourceType`	**Field Type** ExpsSetInterfaceSourceType (enumeration of type string) **Description** The interface source type designed by the consuming cloud that's making a customized expression set builder available to its users. Valid values are: • `Constraint` (Available in API version 62.0 and later). • `DiscoveryProcedure` (Available in API version 61.0 and later). • `EventOrchestration` (Available in API version 61.0 and later). • `PricingProcedure` • `QualificationProcedure` • `RatingDiscoveryProcedure` (Available in API version 61.0 and later). • `Sample` Available in API version 59.0 and later.
`label`	**Field Type** string **Description** Required. The UI label of an expression set definition.
`processType`	**Field Type** ExpsSetProcessType (enumeration of type string) **Description** The process type that uses the expression set rule. Valid values are: • `Bre`

Field Name	Description
	• `GpaCalculation` • `PlanCostCalculation` • `RatingDiscovery` • `StudentSuccess` • Note: When Business Rules Engine is enabled for a Salesforce instance, the default value is '`Bre`'. Other usage types may be available to you depending on your industry solution and permission sets.
`resourceInitializationType`	**Field Type** ResourceInitializationType (enumeration of type string) **Description** Indicates whether the initial value of expression set variables and context tags is null or a default value. Valid values are: • `Default` • `Off` Available in API version 64.0 and later.
`template`	**Field Type** boolean **Description** Defines whether an expression set is a template or not.
`versions`	**Field Type** ExpressionSetDefinitionVersion[] **Description** Represents an array of expression set version definitions in an expression set. Note: This array must contain at least one version.

ExpressionSetDefinitionVersion

Represents a definition of an expression set version.

Field Name	Description
`decimalScale`	**Field Type** integer **Description** Number of decimal places to be used in the results of calculation steps that involve context variables.

Field Name	Description
`description`	**Field Type** string **Description** Describes the version of an expression set definition.
`endDate`	**Field Type** dateTime **Description** The date until which the expression set definition is available for use.
`expressionSetDefinition`	**Field Type** string **Description** The full name of an expression set definition.
`label`	**Field Type** string **Description** Required. The UI label of an expression set definition.
`rank`	**Field Type** int **Description** The rank of the `Expression Set Definition Version`. When more than one enabled version matches an expression set call, and the start date time to end date time spans overlap, the version with the highest rank is chosen. Available in API version 62.0 and later.
`shouldShowExplExternally`	**Field Type** boolean **Description** Indicates whether the decision explanation is exposed to external users (`true`) or not (`false`). The default value is `false`. Available in API version 56.0 and later.
`startDate`	**Field Type** dateTime **Description** Required. The date from when the expression set definition is available for use.

Field Name	Description
`status`	**Field Type** ExpsSetStatus (enumeration of type string) **Description** Required. The status of an expression set definition. Possible values are: • `Active` • `Draft` • `Inactive` • `InvalidDraft` • `Obsolete`
`steps`	**Field Type** ExpressionSetStep[] **Description** Represents an array of steps created in an expression set version.
`uiTier`	**Field Type** boolean **Description** Indicates whether the API call originated from the design time builder or a package. Note: This field is for internal use only.
`variables`	**Field Type** ExpressionSetVariable[] **Description** Represents an array of variables in an expression set version.
`versionNumber`	**Field Type** int **Description** Required. The version number of an expression set definition.

ExpressionSetStep

Represents a step in an expression set version.

Field Name	Description
`actionType`	**Field Type** BusinessKnowledgeModel (enumeration of type string) **Description** Specifies the type of action this step executes. Valid values are: • `AiAcceleratorSubscriberChurnPrediction` • `ApexAction` • `ApexListAction` (Available in API version 64.0 and later.) • `AssetDiscovery` • `AssignBadgeToMember` • `AssignParameterValues` • `AssignmentElement` • `BaseRate` • `BreAggregator` • `BreAggregatorAssignment` • `BreakdownLineMapping` (Available in API version 64.0 and later.) • `ChangeMemberTier` • `CheckMemberBadgeAssignment` • `CommercePricing` (Available in API version 62.0 and later.) • `ComplianceCheck` • `ComplianceControlLog` (Available in API version 62.0 and later.) • `Constraint` (Available in API version 64.0 and later.) • `CreditPoints` • `Crud` • `DebitPoints` • `DerivedPricing` • `DiscountDistributionService` • `DiscoverySettings` (Available in API version 64.0 and later.) • `EvaluateCategoryDisqualification` (Available in API version 62.0 and later.) • `EvaluateCategoryQualification` (Available in API version 62.0 and later.) • `FormulaBasedRating` (Available in API version 62.0 and later.) • `FormulaBasedPricing` • `GetMemberAttributesValues` • `GetMemberPointBalance` • `GetMemberPromotions` • `GetMemberTier` • `GetOutputsFromDecisionMatrix`

Field Name	Description
	• `GetOutputsFromDecisionTable` • `GroupingAndAggregateRating` (Available in API version 62.0 and later.) • `IncreaseUsageForCumulativePromotion` • `IntegrationOrchestration` • `IssueVoucher` • `ManualRatingDiscount` (Available in API version 62.0 and later.) • `MapProduct` • `MinimumPrice` (Available in API version 62.0 and later.) • `MultiRecipientProductQualification` (Available in API version 64.0 and later.) • `NegotiatedBaseRate` (Available in API version 64.0 and later.) • `NegotiatedRateCardEntryResolution` (Available in API version 64.0 and later.) • `NegotiatedTierAdjustment` (Available in API version 64.0 and later.) • `NegotiatedVolumeAdjustment` (Available in API version 64.0 and later.) • `PriceGuidance` (Available in API version 64.0 and later.) • `PricingSettings` • `PromotionsDiscount` • `RateAdjustmentByAttributeResolution` (Available in API version 62.0 and later.) • `RateAdjustmentByTierResolution` (Available in API version 62.0 and later.) • `RateAdjustmentMatrix` (Available in API version 62.0 and later.) • `RateAssignment` (Available in API version 62.0 and later.) • `RateCardEntryResolution` (Available in API version 62.0 and later.) • `RateCardResolution` (Available in API version 62.0 and later.) • `RatingAttributeDiscount` • `RatingRoundingValues` (Available in API version 62.0 and later.) • `RatingSetting` • `RatingTierDiscount` • `RatingVolumeDiscount` • `RecordAction` • `RoundingValues` • `RuleFetch` • `RunFlow` • `RunProgramProcess` • `SampleCustomElementWithExpressionAndListFilter` • `StopPricing` • `StopRating` (Available in API version 62.0 and later.) • `TermGpaCalculation` (Available in API version 64.0 and later.)

Field Name	Description
	• `TermGpaReporting` (Available in API version 64.0 and later.) • `TestCustomElement` • `UpdateCurrentValueForMemberAttribute` • `UpdatePointBalance` • `UpdateUsageForCumulativePromotion` • `UpsertRecord` (Available in API version 64.0 and later.) • `VolumeTierDiscount`
`advancedCondition`	**Field Type** ExpressionSetAdvancedCondition **Description** Represents an advanced condition step.
`aggregation`	**Field Type** ExpressionSetAggregation **Description** Represents an aggregation step.
`assignment`	**Field Type** ExpressionSetAssignment **Description** Represents an assignment step.
`conditionExpression`	**Field Type** ExpressionSetConditionExpression **Description** Represents a condition step.
`customElement`	**Field Type** ExpressionSetCustomElement **Description** Represents a custom element step that contains the input and output mappings. Available in API version 56.0 and later.
`decisionTable`	**Field Type** ExpressionSetDecisionTable **Description** Represents a decision matrix or decision table step.
`description`	**Field Type** string

Field Name	Description
	Description Describes an expression set definition version step.
`failedExplainerTemplate`	**Field Type** string **Description** The explainability message template that's used when the result type of a condition step in an expression set is Failed.
`failedMessageTokenMappings`	**Field Type** ExplainabilityMessageTemplateTokenMapping (enumeration of type string) **Description** List of the token resource mappings of the failed explainability message template. Valid values are: • `expressionSetMessageToken` • `resourceReference` Available in API version 59.0 and later.
`label`	**Field Type** string **Description** Required. The UI label of an expression set definition version step.
`name`	**Field Type** string **Description** Required. The full name of an expression set definition version step.
`noResultExplainerTemplate`	**Field Type** string **Description** The explainability message template that's used when the result type of a condition step in an expression set is No Result. Available in API version 59.0 and later.
`noResultMessageTokenMappings`	**Field Type** ExplainabilityMessageTemplateTokenMapping (enumeration of type string)

Field Name	Description
	Description List of the token resource mappings of the no result explainability message template. Valid values are: • `expressionSetMessageToken` • `resourceReference` Available in API version 59.0 and later.
`parentStep`	**Field Type** string **Description** The name of the parent step in an expression set definition version that's associated with a step.
`passedExplainerTemplate`	**Field Type** string **Description** The explainability message template that's used when the result type of a condition step in an expression set is Passed.
`passedMessageTokenMappings`	**Field Type** ExplainabilityMessageTemplateTokenMapping (enumeration of type string) **Description** List of the token resource mappings of the passed explainability message template. Valid values are: • `expressionSetMessageToken` • `resourceReference` Available in API version 59.0 and later.
`resultIncluded`	**Field Type** boolean **Description** Indicates whether the step output must be included in the expression result (true) or not (false).
`sequenceNumber`	**Field Type** int **Description** Required. The sequence number of a step in an expression set definition version.

Field Name	Description
`shouldExposExecPathMsgOnly`	**Field Type** boolean **Description** Indicates whether the message in the explainability message template is exposed for only the branch path that was run.
`shouldExposeConditionDetails`	**Field Type** boolean **Description** Indicates whether the details of the condition are shown in the decision explanation.
`shouldShowExplExternally`	**Field Type** boolean **Description** Indicates whether the decision explanations are shown to external users.
`stepType`	**Field Type** ExpsSetStepType (enumeration of type string) **Description** Required. Specifies the type of step in an expression set definition version. Valid values are: • `AdvancedCondition` • `Branch` • `BusinessKnowledgeModel` • `Condition` • `DefaultPath` • `SubExpression`
`subExpression`	**Field Type** ExpressionSetSubExpression **Description** Represents a sub expression step.

ExpressionSetAdvancedCondition

Represents an advanced condition step.

Field Name	Description
`conditionLogic`	**Field Type** string **Description** Required. The condition that's defined for an advanced condition.
`criteria`	**Field Type** ExpressionSetConditionCriteria [] **Description** Represents an array of criteria defined in the advanced condition.
`errorMessage`	**Field Type** string **Description** An error message for a failed advanced condition.
`resultParameter`	**Field Type** string **Description** An expression set definition version variable associated with the result of a step.
`successMessage`	**Field Type** string **Description** A success message for a successful advanced condition.

ExpressionSetConditionCriteria

Represents a criterion defined in an advanced condition.

Field Name	Description
`operator`	**Field Type** ExpsSetConditionOperator (enumeration of type string) **Description** Required. Specifies the operator for evaluating an expression. Valid values are: • `Contains` • `DoesNotContain` • `Equals`

Field Name	Description
	• `GreaterThan` • `GreaterThanOrEquals` • `IsNull` • `IsNotNull` • `LessThan` • `LessThanOrEquals` • `NoEquals`
`sequenceNumber`	**Field Type** int **Description** Required. The position of the condition in a step that contains multiple conditions.
`sourceFieldName`	**Field Type** string **Description** Required. The expression set definition version variable associated with the result of a condition criterion.
`value`	**Field Type** string **Description** Specifies the condition of a criterion.
`valueType`	**Field Type** ExpsSetValueType (enumeration of type string) **Description** Specifies the type of value. Valid values are: • `Formula` • `Literal` • `Lookup` • `Parameter` • `Picklist`

ExpressionSetAggregation

Represents an aggregation step.

Field Name	Description
`aggregatedParameter`	**Field Type** string **Description** Required. The expression set definition version variable associated with the result of a condition criterion.
`aggregateFunction`	**Field Type** ExpsSetAggregationFunction (enumeration of type string) **Description** Required. Specifies the aggregation function used in a step. Valid values are: • `Avg` • `Max` • `Min` • `Sum`
`expression`	**Field Type** string **Description** Required. Specifies the expression of an aggregation.

ExpressionSetAssignment

Represents an assignment step.

Field Name	Description
`aggregatedParameter`	**Field Type** string **Description** Required. The expression set definition version variable associated with a step detail.
`expression`	**Field Type** string **Description** Required.

Field Name	Description
	The expression that's defined for a step.

ExpressionSetConditionExpression

Represents a condition in a condition step.

Field Name	Description
`errorMessage`	**Field Type** string **Description** An error message for a failed condition.
`expression`	**Field Type** string **Description** Required. The expression that's defined for a step.
`resultParameter`	**Field Type** string **Description** The expression set definition version variable associated with the result of a step.
`successMessage`	**Field Type** string **Description** A success message for a successful condition.

ExpressionSetCustomElement

Represents a custom element in an expression set. Available in API version 56.0 and later.

Field Name	Description
`parameters`	**Field Type** ExpressionSetElementParameter[] **Description** Represents the list of parameters in the custom element.

ExpressionSetElementParameter

Represents a parameter within a custom element of an expression set. Available in API version 56.0 and later.

Field Name	Description
`input`	**Field Type** boolean **Description** Required. Indicates whether the custom element parameter is input (`true`) or not (`false`). The default value is `true`.
`name`	**Field Type** string **Description** Required. The name of the custom element parameter.
`output`	**Field Type** boolean **Description** Required. Indicates whether the custom element parameter is output (`true`) or not (`false`). The default value is `true`.
`type`	**Field Type** ExpsSetValueType (enumeration of type string) **Description** The type of custom element parameter. Values are: • `Formula` • `Literal` • `Lookup` • `Parameter` • `PickList` The default value is `Parameter`.
`value`	**Field Type** string

Field Name	Description
	Description Required. The name of the expression set variable.

ExpressionSetDecisionTable

Represents a decision table or decision matrix in a step.

Field Name	Description
`decisionTableName`	**Field Type** string **Description** Required. The decision matrix or decision table name used in a step.
`mappings`	**Field Type** ExpressionSetElementParameter[] **Description** The mapping information between various parameters in an ExpressionSetDecisionTable. Available in API version 59.0 and later.
`type`	**Field Type** string **Description** Required. The type in a step. It can be a decision table or decision matrix.

ExpressionSetSubExpression

Represents a sub expression in a step.

Field Name	Description
`expressionSet`	**Field Type** string **Description** Required. The sub expression name used in a step.

Field Name	Description
`mappings`	**Field Type** ExpressionSetElementParameter[] **Description** The mapping information between various parameters in an ExpressionSetDecisionTable. Available in API version 61.0 and later.

ExpressionSetVariable

Represents a definition of an expression set variable.

Field Name	Description
`collection`	**Field Type** boolean **Description** Indicates whether a variable stores a collection of values (`true`) or not (`false`).
`dataType`	**Field Type** ExpsSetDataType (enumeration of type string) **Description** Required. The data type of an expression set variable. Valid values are: • `ActionOutput` • `Boolean` • `Currency` • `Date` • `DateTime` • `DecisionMatrix` • `DecisionTable` • `Numeric` • `Percent` • `Sobject` • `SubExpression` • `Text`
`decimalPlaces`	**Field Type** int

Field Name	Description
	Description The decimal digits in the currency, number, or percent data type for an expression set variable.
`description`	**Field Type** string **Description** The description of the variable used in an expression set.
`fields`	**Field Type** ExpressionSetVariableField [] **Description** Represents an array of fields in an object that is used as a variable in an expression set.
`input`	**Field Type** boolean **Description** Indicates whether an expression set variable is used as an input (`true`) in an expression or not (`false`).
`lookupName`	**Field Type** string **Description** The API name of a decision matrix, a decision table, or a sub expression.
`lookupType`	**Field Type** ExpsSetVariableLookupType (enumeration of type string) **Description** The type of the lookup used in an expression set definition. Valid values are: • `DecisionMatrix` • `DecisionTable` • `SubExpression`
`name`	**Field Type** string **Description** Required. The full name of the variable used in an expression set definition.

Field Name	Description
`objectName`	**Field Type** string **Description** The name of the sObject.
`output`	**Field Type** boolean **Description** Indicates whether an expression set variable is used as an output in an expression(`true`) or not (`false`).
`resultStep`	**Field Type** string **Description** The step that produces the expression set variable.
`type`	**Field Type** ExpsSetVariableType (enumeration of type string) **Description** Required. The type of variable in an expression set definition. Valid values are: • `Constant` • `ContextDynamicAttributeTag` (Available in API version 62.0 and later.) • `ExecutableContextDefinitionTag` (Available in API version 62.0 and later.) • `Formula` • `Variable`
`value`	**Field Type** string **Description** Represents a constant value or a formula. Note: It stores the default value of a variable.

ExpressionSetVariableField

Represents a definition of a field in an object that is used as a variable in an expression set.

Field Name	Description
`dataType`	**Field Type** ExpsSetDataType (enumeration of type string) **Description** Required. Specifies the type of data stored in an expression set variable. Valid values are: • `ActionOutput` • `Boolean` • `Currency` • `Date` • `DateTime` • `DecisionMatrix` • `DecisionTable` • `Numeric` • `Percent` • `Sobject` • `SubExpression` • `Text`
`decimalPlaces`	**Field Type** int **Description** The decimal digits in the currency, number, or percent data type for an expression set variable.
`fields`	**Field Type** ExpressionSetVariableField [] **Description** Represents an array of fields in an object that is used as a variable in an expression set.
`lookupName`	**Field Type** string **Description** The API name of a decision matrix, a decision table, or a sub expression.
`lookupType`	**Field Type** ExpsSetVariableLookupType (enumeration of type string) **Description** Required. The type of lookup used in an expression set definition.

Field Name	Description
	Valid values are: • `DecisionMatrix` • `DecisionTable` • `SubExpression`
`name`	**Field Type** string **Description** Required. The full name of the field used in an expression set variable.
`objectName`	**Field Type** string **Description** The name of the sObject.

Declarative Metadata Sample Definition

The following is an example of an ExpressionSetDefinition component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ExpressionSetDefinition xmlns="http://soap.sforce.com/2006/04/metadata">
    <label>ExpSetWithAllSteps</label>
    <processType>Bre</processType>
    <template>false</template>
    <description></description>
    <interfaceSourceType>Sample</interfaceSourceType>
    <executionScale>Low</executionScale>
    <versions>
        <fullName>ExpSetWithAllSteps_V1</fullName>
        <expressionSetDefinition>ExpSetWithAllSteps</expressionSetDefinition>
        <label>ExpSetWithAllSteps V1</label>
        <shouldShowExplExternally>false</shouldShowExplExternally>
        <startDate>2022-08-09T22:04:56.000Z</startDate>
        <endDate>2023-08-09T22:04:56.000Z</endDate>
        <status>Draft</status>
        <uiTier>false</uiTier>
        <rank>1</rank>
        <description>ExpSetWithAllSteps_V1</description>
        <steps>
            <description>Aggregate</description>
            <actionType>BreAggregator</actionType>
            <aggregation>
                <aggergatedParameter>result</aggergatedParameter>
                <aggregateFunction>Avg</aggregateFunction>
                <expression>AVG ( result )</expression>
            </aggregation>
```

```
        <label>Aggregate</label>
        <name>Aggregate</name>
        <resultIncluded>true</resultIncluded>
        <sequenceNumber>5</sequenceNumber>
        <shouldExposExecPathMsgOnly>true</shouldExposExecPathMsgOnly>
        <shouldExposeConditionDetails>false</shouldExposeConditionDetails>
        <shouldShowExplExternally>false</shouldShowExplExternally>
        <stepType>BusinessKnowledgeModel</stepType>
    </steps>
    <steps>
        <label>Branch</label>
        <name>Branch</name>
        <resultIncluded>false</resultIncluded>
        <sequenceNumber>4</sequenceNumber>
        <shouldExposExecPathMsgOnly>true</shouldExposExecPathMsgOnly>
        <shouldExposeConditionDetails>false</shouldExposeConditionDetails>
        <shouldShowExplExternally>false</shouldShowExplExternally>
        <stepType>Branch</stepType>
    </steps>
    <steps>
        <actionType>AssignParameterValues</actionType>
        <assignment>
            <assignedParameter>b</assignedParameter>
            <expression>SUM ( a , 10 )</expression>
        </assignment>
        <label>Calculation</label>
        <name>Calculation</name>
        <resultIncluded>true</resultIncluded>
        <sequenceNumber>1</sequenceNumber>
        <shouldExposExecPathMsgOnly>true</shouldExposExecPathMsgOnly>
        <shouldExposeConditionDetails>false</shouldExposeConditionDetails>
        <shouldShowExplExternally>false</shouldShowExplExternally>
        <stepType>BusinessKnowledgeModel</stepType>
    </steps>
    <steps>
        <actionType>AssignParameterValues</actionType>
        <assignment>
            <assignedParameter>result</assignedParameter>
            <expression>b * 100</expression>
        </assignment>
        <label>Calculation</label>
        <name>Calculation10</name>
        <parentStep>DefaultLane</parentStep>
        <resultIncluded>false</resultIncluded>
        <sequenceNumber>1</sequenceNumber>
        <shouldExposExecPathMsgOnly>true</shouldExposExecPathMsgOnly>
        <shouldExposeConditionDetails>false</shouldExposeConditionDetails>
        <shouldShowExplExternally>false</shouldShowExplExternally>
        <stepType>BusinessKnowledgeModel</stepType>
    </steps>
    <steps>
        <actionType>AssignParameterValues</actionType>
        <assignment>
            <assignedParameter>result</assignedParameter>
```

```
            <expression>b * 1</expression>
        </assignment>
        <label>Calculation</label>
        <name>Calculation3</name>
        <parentStep>Condition</parentStep>
        <resultIncluded>false</resultIncluded>
        <sequenceNumber>1</sequenceNumber>
        <shouldExposExecPathMsgOnly>true</shouldExposExecPathMsgOnly>
        <shouldExposeConditionDetails>false</shouldExposeConditionDetails>
        <shouldShowExplExternally>false</shouldShowExplExternally>
        <stepType>BusinessKnowledgeModel</stepType>
    </steps>
    <steps>
        <actionType>AssignParameterValues</actionType>
        <assignment>
            <assignedParameter>result</assignedParameter>
            <expression>SUM ( b , 10 )</expression>
        </assignment>
        <label>Calculation</label>
        <name>Calculation5</name>
        <parentStep>Condition4</parentStep>
        <resultIncluded>false</resultIncluded>
        <sequenceNumber>1</sequenceNumber>
        <shouldExposExecPathMsgOnly>true</shouldExposExecPathMsgOnly>
        <shouldExposeConditionDetails>false</shouldExposeConditionDetails>
        <shouldShowExplExternally>false</shouldShowExplExternally>
        <stepType>BusinessKnowledgeModel</stepType>
    </steps>
    <steps>
        <actionType>AssignParameterValues</actionType>
        <assignment>
            <assignedParameter>result</assignedParameter>
            <expression>b * 10</expression>
        </assignment>
        <label>Calculation</label>
        <name>Calculation8</name>
        <parentStep>Condition7</parentStep>
        <resultIncluded>false</resultIncluded>
        <sequenceNumber>1</sequenceNumber>
        <shouldExposExecPathMsgOnly>true</shouldExposExecPathMsgOnly>
        <shouldExposeConditionDetails>false</shouldExposeConditionDetails>
        <shouldShowExplExternally>false</shouldShowExplExternally>
        <stepType>BusinessKnowledgeModel</stepType>
    </steps>
    <steps>
        <conditionExpression>
            <successMessage>success</successMessage>
            <errorMessage>error</errorMessage>
            <expression>IS10 == b</expression>
            <resultParameter>condition_output__1</resultParameter>
        </conditionExpression>
        <label>Condition</label>
        <name>Condition</name>
        <resultIncluded>false</resultIncluded>
```

```
        <sequenceNumber>2</sequenceNumber>
        <shouldExposExecPathMsgOnly>true</shouldExposExecPathMsgOnly>
        <shouldExposeConditionDetails>false</shouldExposeConditionDetails>
        <shouldShowExplExternally>false</shouldShowExplExternally>
        <stepType>Condition</stepType>
    </steps>
    <steps>
        <advancedCondition>
            <successMessage>success</successMessage>
            <errorMessage>error</errorMessage>
            <conditionLogic>1</conditionLogic>
            <criteria>
                <operator>Equals</operator>
                <sequenceNumber>1</sequenceNumber>
                <sourceFieldName>condition_output__1</sourceFieldName>
                <value>true</value>
                <valueType>Literal</valueType>
            </criteria>
            <resultParameter>condition_output__3</resultParameter>
        </advancedCondition>
        <label>Condition</label>
        <name>Condition4</name>
        <resultIncluded>false</resultIncluded>
        <sequenceNumber>3</sequenceNumber>
        <shouldExposExecPathMsgOnly>true</shouldExposExecPathMsgOnly>
        <shouldExposeConditionDetails>false</shouldExposeConditionDetails>
        <shouldShowExplExternally>false</shouldShowExplExternally>
        <stepType>AdvancedCondition</stepType>
    </steps>
    <steps>
        <conditionExpression>
            <expression>IS10 == b</expression>
            <resultParameter>condition_output__2</resultParameter>
        </conditionExpression>
        <label>Condition</label>
        <name>Condition7</name>
        <parentStep>Branch</parentStep>
        <resultIncluded>false</resultIncluded>
        <sequenceNumber>1</sequenceNumber>
        <shouldExposExecPathMsgOnly>true</shouldExposExecPathMsgOnly>
        <shouldExposeConditionDetails>false</shouldExposeConditionDetails>
        <shouldShowExplExternally>false</shouldShowExplExternally>
        <stepType>Condition</stepType>
    </steps>
    <steps>
        <label>Default Lane</label>
        <name>DefaultLane</name>
        <parentStep>Branch</parentStep>
        <resultIncluded>false</resultIncluded>
        <sequenceNumber>2</sequenceNumber>
        <shouldExposExecPathMsgOnly>true</shouldExposExecPathMsgOnly>
        <shouldExposeConditionDetails>false</shouldExposeConditionDetails>
        <shouldShowExplExternally>false</shouldShowExplExternally>
        <stepType>DefaultPath</stepType>
```

```
    </steps>
    <steps>
        <actionType>AssignParameterValues</actionType>
        <assignment>
            <assignedParameter>a</assignedParameter>
            <expression>3</expression>
        </assignment>
        <failedExplainerTemplate>CalculationFailure</failedExplainerTemplate>
        <failedMessageTokenMappings>
            <expressionSetMessageToken>y2</expressionSetMessageToken>
            <resourceReference>a</resourceReference>
        </failedMessageTokenMappings>
        <label>CalculationStepWithTokensAndMappings</label>
        <name>CalculationStepWithTokensAndMappings</name>
        <passedExplainerTemplate>CalculationSuccess</passedExplainerTemplate>
        <passedMessageTokenMappings>
            <expressionSetMessageToken>y1</expressionSetMessageToken>
            <resourceReference>a</resourceReference>
        </passedMessageTokenMappings>
        <resultIncluded>false</resultIncluded>
        <sequenceNumber>1</sequenceNumber>
        <shouldExposExecPathMsgOnly>true</shouldExposExecPathMsgOnly>
        <shouldExposeConditionDetails>false</shouldExposeConditionDetails>
        <shouldShowExplExternally>true</shouldShowExplExternally>
        <stepType>BusinessKnowledgeModel</stepType>
    </steps>
    <variables>
        <collection>false</collection>
        <dataType>Boolean</dataType>
        <description>condition_output__3</description>
        <input>false</input>
        <name>condition_output__3</name>
        <output>false</output>
        <resultStep>Condition4</resultStep>
        <type>Variable</type>
        <value>False</value>
    </variables>
    <variables>
        <collection>false</collection>
        <dataType>Numeric</dataType>
        <decimalPlaces>2</decimalPlaces>
        <description>a</description>
        <input>true</input>
        <name>a</name>
        <output>false</output>
        <type>Variable</type>
        <value>10</value>
    </variables>
    <variables>
        <collection>false</collection>
        <dataType>Boolean</dataType>
        <description>condition_output__1</description>
        <input>false</input>
        <name>condition_output__1</name>
```

```
            <output>false</output>
            <resultStep>Condition</resultStep>
            <type>Variable</type>
            <value>False</value>
        </variables>
        <variables>
            <collection>false</collection>
            <dataType>Boolean</dataType>
            <description>condition_output__2</description>
            <input>false</input>
            <name>condition_output__2</name>
            <output>false</output>
            <resultStep>Condition7</resultStep>
            <type>Variable</type>
            <value>False</value>
        </variables>
        <variables>
            <collection>false</collection>
            <dataType>Numeric</dataType>
            <decimalPlaces>2</decimalPlaces>
            <description>IS10</description>
            <input>false</input>
            <name>IS10</name>
            <output>false</output>
            <type>Constant</type>
            <value>10</value>
        </variables>
        <variables>
            <collection>false</collection>
            <dataType>Numeric</dataType>
            <decimalPlaces>2</decimalPlaces>
            <description>b</description>
            <input>false</input>
            <name>b</name>
            <output>true</output>
            <type>Variable</type>
        </variables>
        <variables>
            <collection>false</collection>
            <dataType>Numeric</dataType>
            <decimalPlaces>2</decimalPlaces>
            <description>result</description>
            <input>false</input>
            <name>result</name>
            <output>true</output>
            <type>Variable</type>
        </variables>
        <versionNumber>1</versionNumber>
    </versions>
</ExpressionSetDefinition>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package
```

```
 xmlns="http://soap.sforce.com/2006/04/metadata">
 <types>
  <members>*</members>
  <name>ExpressionSetDefinition</name>
 </types>
 <version>64.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ExpressionSetMessageToken

Represents an interface to retrieve, deploy, create, update, or delete information on Expression Set Message Token.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

ExpressionSetMessageToken components have the suffix `expressionSetMessageToken` and are stored in the `ExpressionSetMessageToken` folder.

Version

ExpressionSetMessageToken components are available in API version 59.0 and later.

Special Access Rules

InteractionCalculation.orgHasBREandDESAccess Org permission set license is required for users to access this metadata type.

Fields

Field Name	Description
`description`	**Field Type** string **Description** Required.

Field Name	Description
	Description of the expression set message token.
`developerName`	**Field Type** string **Description** Required. Developer name of the expression set message token.
`masterLabel`	**Field Type** string **Description** Required. A user-friendly name for ExpressionSetMessageToken, which is defined when the ExpressionSetMessageToken is created.

Declarative Metadata Sample Definition

The following is an example of an ExpressionSetMessageToken component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ExpressionSetMessageToken xmlns="http://soap.sforce.com/2006/04/metadata">
    <developerName>token</developerName>
    <description>Description</description>
    <masterLabel>token</masterLabel>
</ExpressionSetMessageToken>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>ExpressionSetMessageToken</name>
    </types>
    <version>59.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ExpressionSetObjectAlias

Represents information about the alias of the source object that's used in an expression set.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

ExpressionSetObjectAlias components have the suffix `.expressionSetObjectAlias` and are stored in the `expressionSetObjectAlias` folder.

Version

ExpressionSetObjectAlias components are available in API version 56.0 and later.

Fields

Field Name	Description
`dataType`	**Field Type** ExpsSetObjectDataType (enumeration of type string) **Description** Required. The data type of the object alias. Values are: • `JSON` • `sObject`
`mappings`	**Field Type** ExpressionSetObjectAliasField[] **Description** The mapping between a source field and its corresponding field alias.
`objectApiName`	**Field Type** string **Description** Required. The API name of the top-level object, when the data type is sObject. The key of the top-level object, when the data type is JSON.
`usageType`	**Field Type** ExpsSetProcessType (enumeration of type string) **Description** Required.

Field Name	Description
	The type of application associated with the industry that's using an expression set. Your Salesforce org admin can define the values. Valid values are: • `Bre` • Note: When Business Rules Engine is enabled for a Salesforce instance, the default value is '`Bre`'. Other usage types may be available to you depending on your industry solution and permission sets.

ExpressionSetObjectAliasField

The fields associated with the source object for which the object alias is created.

Field Name	Description
`fieldAlias`	**Field Type** string **Description** Required. The field alias associated with the source field name.
`sourceFieldName`	**Field Type** string **Description** Required. The name of the source field for which the field alias is created. The source field name under an object alias must be unique.

Declarative Metadata Sample Definition

The following is an example of an ExpressionSetObjectAlias component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ExpressionSetObjectAlias xmlns="http://soap.sforce.com/2006/04/metadata">
    <dataType>sObject</dataType>
    <mappings>
        <fieldAlias>dum2</fieldAlias>
        <sourceFieldName>CreatedBy.Contact.Name</sourceFieldName>
    </mappings>
    <mappings>
        <fieldAlias>dum3</fieldAlias>
        <sourceFieldName>CreatedBy.Name</sourceFieldName>
    </mappings>
```

```
    <mappings>
        <fieldAlias>dum1</fieldAlias>
        <sourceFieldName>Owner.Contact.Name</sourceFieldName>
    </mappings>
    <objectApiName>Account</objectApiName>
    <usageType>Bre</usageType>
</ExpressionSetObjectAlias>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package
 xmlns="http://soap.sforce.com/2006/04/metadata">
 <types>
  <members>*</members>
  <name>ExpressionSetObjectAlias</name>
 </types>
 <version>64.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ExternalAuthIdentityProvider

Represents an external authentication (auth) identity provider. An external auth identity provider links to an external credential and obtains OAuth tokens for outbound callouts to external systems.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

ExternalAuthIdentityProvider components have the suffix `.externalAuthIdentityProvider` and are stored in the `externalAuthIdentityProviders` folder.

Version

ExternalAuthIdentityProvider components are available in API version 62.0 and later.

Special Access Rules

Only users with the Customize Application permission or the Manage Named Credentials permission can access this type.

Fields

Field Name	Description
`authenticationFlow`	**Field Type** IdentityProviderAuthFlow (enumeration of type string) **Description** Required. Authentication flow to get tokens to call protected APIs. Values are: • `AuthorizationCode` • `SalesforceDefined`
`authenticationProtocol`	**Field Type** IdentityProviderAuthProtocol (enumeration of type string) **Description** Required. The authentication protocol that's required to access the external system. Values are: • `OAuth` • `SalesforceDefined`
`description`	**Field Type** string **Description** A meaningful description of the external auth identity provider.
`externalAuthIdentityProviderParameters`	**Field Type** ExternalAuthIdentityProviderParameter[] **Description** One or more sets of parameters that further configure the external auth identity provider.
`label`	**Field Type** string **Description** Required. Name of the external auth identity provider.

ExternalAuthIdentityProviderParameter

Represents the parameters that configure an external auth identity provider.

These parameters are used internally to provide a flexible architecture and are exposed here for packaging reasons.

Field Name	Description
`description`	**Field Type** string **Description** A human-readable description of this external auth identity provider parameter.
`parameterName`	**Field Type** string **Description** Required. The name of the external auth identity provider parameter.
`parameterType`	**Field Type** ExtIIdentityProviderParmType (enumeration of type string) **Description** Required. The type of external auth identity provider parameter. The value of this field drives the behavior of the parameter. Values are: • `AuthorizeRequestQueryParameter` • `AuthorizeUrl` • `ClientAuthentication` • `CreatedByNamespace` • `IdentityProviderOptions` • `RefreshRequestBodyParameter` • `RefreshRequestHttpHeader` • `RefreshRequestQueryParameter` • `StandardExternalIdentityProvider` • `TokenRequestBodyParameter` • `TokenRequestHttpHeader` • `TokenRequestQueryParameter` • `TokenUrl` • `UserInfoUrl`
`parameterValue`	**Field Type** string **Description** If the `parameterType` field describes a literal value then the literal value is stored in this field.
`sequenceNumber`	**Field Type** int

Field Name	Description
	Description Specifies the order of parameters to apply when an external auth identity provider has more than one parameter. Priority is from lower to higher numbers (for example, 1 is the highest priority).

Declarative Metadata Sample Definition

The following is an example of an ExternalAuthIdentityProvider component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ExternalAuthIdentityProvider xmlns="http://soap.sforce.com/2006/04/metadata">
    <authenticationFlow>AuthorizationCode</authenticationFlow>
    <authenticationProtocol>OAuth</authenticationProtocol>
    <description>OAuth Browser flow for connected app</description>
    <externalAuthIdentityProviderParameter>
        <parameterName>TokenUrl</parameterName>
        <parameterType>TokenUrl</parameterType>
        <parameterValue>https://localhost:6101/services/oauth2/token</parameterValue>
        <sequenceNumber>1</sequenceNumber>
    </externalAuthIdentityProviderParameter>
    <externalAuthIdentityProviderParameter>
        <parameterName>AuthorizeUrl</parameterName>
        <parameterType>AuthorizeUrl</parameterType>
        <parameterValue>https://localhost:6101/services/oauth2/authorize</parameterValue>

        <sequenceNumber>2</sequenceNumber>
    </externalAuthIdentityProviderParameter>
    <label>exampleExtlIdp</label>
</ExternalAuthIdentityProvider>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>ExternalAuthIdentityProvider</name>
    </types>
    <version>62.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ExternalClientApplication

Represents the header file for an external client application configuration.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

ExternalClientApplication components have the suffix `.eca` and are stored in the `externalClientApps` folder.

Version

ExternalClientApplication components are available in API version 59.0 and later.

Special Access Rules

Access to the ExternalClientApplication type requires orgs to enable the Opt in to External Client Apps permission in Setup.

Fields

Field Name	Description
`contactEmail`	**Field Type** string **Description** The email address that Salesforce uses to contact the external client app admin for the subscriber org.
`contactPhone`	**Field Type** string **Description** The phone number that Salesforce uses to contact the external client app admin for the subscriber org.
`description`	**Field Type** string **Description** A description for the app.
`distributionState`	**Field Type** ExtlClntAppDistState (enumeration of type string)

Field Name	Description
	Description The distribution state of an external client app. Values are: • `AutoInstalled`. For internal use only. • `Local`. • `Managed`. For internal use only. • `Packaged`.
`iconUrl`	**Field Type** string **Description** The URL for the icon image.
`infoUrl`	**Field Type** string **Description** Reserved for future use.
`isProtected`	**Field Type** boolean **Description** A package construct that developers use to control the visibility of components in subscriber orgs. Default is false.
`label`	**Field Type** string **Description** The label for the external client app.
`logoUrl`	**Field Type** string **Description** The URL for the logo image.
`managedType`	**Field Type** ExtlClntAppManagedType (enumeration of type string) **Description** For internal use only.
`orgScopedExternalApp`	**Field Type** string

Field Name	Description
	Description A unique ID consisting of the org ID and the name of this external client app. Either defined by the developer or auto-generated during the first deployment. The expected value uses this format: *[Organization_ID]:[External Client App Name]*.

Declarative Metadata Sample Definition

This example shows an ExternalClientApplication component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ExternalClientApplication xmlns="http://soap.sforce.com/2006/04/metadata">
    <contactEmail>johndoe@example.com</contactEmail>
    <description>Test external client app</description>
    <distributionState>Local</distributionState>
    <iconUrl>https://icon.example.com</iconUrl>
    <infoUrl>https://info.example.com</infoUrl>
    <logoUrl>https://logo.example.com</logoUrl>
    <label>myeca</label>
    <isProtected>false</isProtected>
    <orgScopedExternalApp>Org_ID:External_Client_App_Name</orgScopedExternalApp>
</ExternalClientApplication>
```

This example `package.xml` references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>ExternalClientApplication</name>
    </types>
    <types>
        <members>*</members>
        <name>ExtlClntAppOauthSettings</name>
    </types>
    <types>
        <members>*</members>
        <name>ExtlClntAppGlobalOauthSettings</name>
    </types>
    <types>
        <members>*</members>
        <name>ExtlClntAppOauthConfigurablePolicies</name>
    </types>
    <types>
        <members>*</members>
        <name>ExtlClntAppConfigurablePolicies</name>
    </types>
    <version>60.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ExternalCredential

Represents the details of how Salesforce authenticates to the external system.

Note: All credentials stored within this entity are encrypted under a framework that is consistent with other encryption frameworks on the platform. Salesforce encrypts your credentials by auto-creating org-specific keys. Credentials encrypted using the previous encryption scheme have been migrated to the new framework.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

ExternalCredential components have the suffix `.externalCredential` and are stored in the `externalCredentials` folder.

Version

ExternalCredential components are available in API version 56.0 and later.

Special Access Rules

There are no additional access requirements that are specific to this type.

Fields

Field Name	Description
`authenticationProtocol`	**Field Type** AuthenticationProtocol (enumeration of type string) **Description** Required. The authentication protocol that's required to access the external system. Valid values are: • `AwsSv4` • `Basic` • `Custom` — User-created authentication. Specify the permission set, sequence number, and authentication parameters. Each authentication parameter requires a name and value.

Field Name	Description
	• `Jwt` — Reserved for future use • `JwtExchange` — Reserved for future use • `NoAuthentication` —Reserved for future use • `Oauth` • `Password` — Reserved for future use For connections to Amazon Web Services using Signature Version 4, use `AwsSv4`. For connections using a direct token system, select `Jwt`. For Simple URL data sources, select `Custom` with no parameters. For cloud-based Files Connect external systems, select `Oauth`. For on-premises systems, select `Password`.
`description`	**Field Type** string **Description** A meaningful description of the external credential.
`externalCredentialParameters`	**Field Type** ExternalCredentialParameter[] **Description** One or more sets of parameters that further configure the external credential.
`label`	**Field Type** string **Description** Required. Name of the external credential.

ExternalCredentialParameter

Represents the parameters that configure an external credential. External credential parameters are used to configure external credential callouts through a combination of the type, name, and value and lookup fields. Available in API version 56.0 and later.

These parameters are used internally to provide a flexible architecture and are exposed here for packaging reasons.

Field Name	Description
`authProvider`	**Field Type** string

Field Name	Description
	Description Reference to an authentication provider that the `AuthProvider` component represents, which defines the service that provides the login process and approves access to the external system.
`certificate`	**Type** string **Description** If the value of `parameterType` is `SigningCertificate`, then this field references the certificate.
`description`	**Field Type** string **Description** A human-readable description of this external credential parameter.
`externalAuthIdentityProvider`	**Field Type** string **Description** Reference to an external authentication identity provider that the `externalAuthIdentityProvider` component represents. The `externalAuthIdentityProvider` defines the service that provides the login process and approves access to the external system. To simplify the configuration process for the authentication providers used by your named credentials, use an `externalAuthIdentityProvider` instead of an `authProvider`. Link the external auth identity provider to an external credential.
`parameterGroup`	**Field Type** string **Description** Groups a parameter along with its respective principal. For example, with dynamic scopes, the user can apply a scope `AuthParameter` only when authenticated against a specific principal with a matching `parameterGroup` value. If a value for `parameterGroup` isn't provided, `parameterGroup` defaults to the `parameterName` value for PER_USER and NAMED_PRINCIPAL. For all other parameters `parameterGroup` defaults to DEFAULT_GROUP.
`parameterName`	**Field Type** string **Description** Required. The name of the external credential parameter.

Field Name	Description
`parameterType`	**Field Type** ExternalCredentialParamType (enumeration of type string) **Description** Required. The type of external credential parameter. The value of this field drives the behavior of the parameter. Valid values are: • `AdditionalRefreshStatusCode`: Allows the user to specify 4*xx*, 6*xx*, 7*xx*, 8*xx*, and 9*xx* HTTP status codes that trigger Salesforce to refresh expired or invalid access tokens, in addition to the standard `401` HTTP status code response. • `AuthHeader`: Allows the user to specify custom authentication headers to be added to the callout at run time. When using `AuthHeader`, the `parameterName` field must be the header name as a string, and `parameterValue` must be a formula of a header value that is evaluated at run time. `sequenceNumber` determines the order in which headers are sent out in the callout. Headers with lower numbers are sent out first. • `AuthParameter`: Allows the user to add additional authentication settings. `parameterName` defines the parameter to set. For example, `AwsRegion` sets the AWS Region parameter to apply for an AWS Signature V4 authentication protocol and `parameterValue` is the value for the AWS Region. • `AuthProtocolVariant`: Used to specify a variant of an authentication protocol. For example, `Aws Sts` as a variant when the `ParameterName` is `AwsSv4` and the `ParameterValue` is `AwsSv4_STS`. • `AuthProvider`: Specifies that this parameter configures an authentication provider referenced by the `authProvider` field. • `AuthProviderUrl`: Specifies the authentication endpoint URL. For example, if the authentication type is OAuth with JWT Bearer Flow, then `parameterValue` is an authentication token endpoint. • `AuthProviderUrlQueryParameter`: Allows the user to specify custom query parameters to be added to the callout to the authentication provider at run time. Currently, supported only for AWS Signature V4 with STS. The allowed `AuthProviderUrlQueryParameter` values are `AwsExternalId` and `AwsDuration`, used with AWS STS. • `AwsStsPrincipal`: Configures AWS Signature V4 along with STS. `parameterName` is `AwsStsPrincipal` and `parameterValue` isn't specified. • `CreatedByNamespace`: Reserved for internal use. • `ExternalAuthIdentityProvider`: Specifies that this parameter configures an authentication provider referenced by the `externalAuthIdentityProvider` field. • `GlobalNamedPrincipal`: Reserved for internal use.

Field Name	Description
	• `JwtBodyClaim`: Specifies a JWT (JSON Web Token) body claim, where `parameterName` is the key and `parameterValue` is the value. For example, the parameter name for a JWT audience is `aud`. • `JwtHeaderClaim`: Specifies a JWT header claim, where `parameterName` is the key and `parameterValue` is the value. For example, the parameter name for a JWT key identifier is `kid`. • `NamedPrincipal`: Specifies that the parameter uses the same set of user credentials for all users who access the external system. • `PerUserPrincipal`: Provides access control at the individual user level. • `SigningCertificate`: Specifies the certificate used for an authentication signature. Use the `certificate` field to specify the certificate name. Used for OAuth with JWT Bearer Flow and AwsSv4 STS with RolesAnywhere authentication. • `SystemUserPrincipal`: Reserved for internal use.
`parameterValue`	**Field Type** string **Description** If the `parameterType` field describes a literal value then the literal value is stored in this field.
`principal`	**Field Type** string **Description** If the value of the `parameterType` field is either `NamedPrincipal` or `PerUserPrincipal`, this field points to a permission set. That value then determines the set of users that are allowed to use credentials provided by the credential provider. The value of the `parameterName` field specifies the name of this principal. **First available in API version 56.0, this field is removed in API version 58.0 and later.**
`sequenceNumber`	**Field Type** int **Description** Specifies the order of principals to apply when a user participates in more than one principal. For example, a user could be part of multiple permission sets that are applicable for a credential provider. Priority is from lower to higher numbers. You can set this field only when `parameterType` is `NamedPrincipal`.

Declarative Metadata Sample Definition

The following is an example of an ExternalCredential component.

```
<ExternalCredential xmlns="http://soap.sforce.com/2006/04/metadata">
    <label>SampleExternalCredential</label>
    <authenticationProtocol>AwsSv4</authenticationProtocol>
    <externalCredentialParameters>
        <parameterName>Principal</parameterName>
        <parameterType>NamedPrincipal</parameterType>
        <sequenceNumber>1</sequenceNumber>
    </externalCredentialParameters>
    <externalCredentialParameters>
        <parameterName>AwsService</parameterName>
        <parameterValue>iam</parameterValue>
        <parameterType>AuthParameter</parameterType>
    </externalCredentialParameters>
    <externalCredentialParameters>
        <parameterName>AwsRegion</parameterName>
        <parameterValue>us-east-1</parameterValue>
        <parameterType>AuthParameter</parameterType>
    </externalCredentialParameters>
</ExternalCredential>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>ExternalCredential</name>
    </types>
    <version>56.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SEE ALSO:

- NamedCredential
- *Salesforce Help*: Named Credentials
- *Named Credentials Developer Guide*: Get Started with Named Credentials
- *Named Credentials Developer Guide*: Named Credential API Links
- *Apex Developer Guide*: Invoking Callouts Using Apex
- *Apex Developer Guide*: Named Credentials as Callout Endpoints

ExternalAIModel

Represents the state of a given model for an Einstein for Service feature, such as Einstein Reply Recommendations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

ExternalAIModel components have the suffix `.externalAIModel` and are stored in the `externalAIModels` folder.

Version

ExternalAIModel components are available in API version 51.0 and later.

Special Access Rules

This type is available only when an org is configured to access the application in the `applicationSourceType` field. For example, if `applicationSourceType` is set to `ARTICLE_RECOMMENDATION`, this type is available only if Einstein Article Recommendations is enabled in the org and the Main Services Agreement has been accepted.

Fields

Field Name	Field Type	Description
`applicationSourceType`	ApplicationSourceType (enumeration of type string)	Required. The target application for the configuration. Valid values are: • `REPLY_RECOMMENDATION`— Einstein Reply Recommendations • `ARTICLE_RECOMMENDATION`— Einstein Article Recommendations • `UTTERANCE_RECOMMENDATION`— Einstein Bot utterances • `FAQ`— Einstein Bot frequently asked questions
`externalModelKey`	string	Required. Unique key which identifies external model corresponding this applicationType
`externalModelStatus`	ExternalModelStatus (enumeration of type string)	Required. The current state of a given model. Valid values are: • `DISABLED` • `ENABLED` • `PAUSED`
`name`	string	Required. A reference to the configuration.
`threshold`	double	Threshold override value for this model. Nillable.

Field Name	Field Type	Description
trainingJobName	string	Training job path corresponding to the given model. Nillable.

Declarative Metadata Sample Definition

The following is an example of an ExternalAIModel component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ExternalAIModel xmlns="http://soap.sforce.com/2006/04/metadata">
    <applicationSourceType>REPLY_RECOMMENDATION</applicationSourceType>
    <externalModelKey>0f16dea6-b886-44df-9cfa-4d96b51d6594</externalModelKey>
    <externalModelStatus>ENABLED</externalModelStatus>
    <name>SR1601228426202</name>
    <threshold>0.9</threshold>
    <trainingJobName>TestJob</trainingJobName>
</ExternalAIModel>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>ExternalAIModel</name>
    </types>
</Package>
```

Wildcard Support in the Manifest File

This metadata type doesn't support the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ExternalServiceRegistration

Represents the External Service configuration for an org. This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

ExternalServiceRegistration components have the suffix `.externalServiceRegistration` and are stored in the `externalServiceRegistrations` folder.

Version

ExternalServiceRegistration components are available in API version 39.0 and later.

Fields

Field Name	Field Type	Description
`description`	string	The external service description defined when the service is created.
`label`	string	Required. The service name as it appears on the External Services wizard.
`namedCredential`	string	The reference by name to be used for the service.
`NamedCredentialReferenceId`	reference	The reference by ID to be used for the named credential. When used, supersedes `namedCredential`. Available in API version 57.0 and later.
`operations`	ExternalServiceOperation[]	Items defined for this operation.
`registrationProvider`	string	A reference to the registration provider. • If the registrationProviderType is `ExternalConnector`, this field contains the external connector name. • If the registrationProviderType is `Heroku`, this field contains the HerokuAppLink ID. • For any other registrationProviderType value, this field is blank and reserved for future use.
`registrationProviderType`	ExternalServiceRegistrationProviderType (enumeration of type string)	Indicates the source of the API specification registered with the External Services wizard. Valid values include: • `Anypoint`—The API spec was defined in MuleSoft Anypoint. Available in API version 63.0 and later. • `ApexRest`—The API spec was created from an Apex REST class. Available in API version 63.0 and later. • `Custom`—The API spec was manually configured. • `ExternalConnector`—The API spec represents an external connection. • `Heroku`—The API spec represents a Heroku app. • `MuleSoft`—The API spec was selected from MuleSoft. • `SchemaInferred`—The API spec was provided during the HTTP Callout configuration process. Available in API version 57.0 and later. • `Standard`—The API spec was defined when an external service was created.
`schema`	string	The content of the OpenAPI 2.0.x or OpenAPI 3.0.x schema in JSON or YAML format. Nillable.
`schemaAbsoluteUrl`	string	The full, absolute URL to the schema. Populated when a user selects **Absolute URL** during registration.
`schemaType`	string	The schema format. OpenAPI for Open API 2.0.x or Open API 3.0.x. If not specified, schema type is derived based on the schema content. Nillable.

Field Name	Field Type	Description
`schemaUploadFileExtension`	string	The file's extension. Populated when a user selects **Upload from local** during registration.
`schemaUploadFileName`	string	The file's name without the file extension. Populated when a user selects **Upload from local** during registration.
`schemaUrl`	string	The path should begin with "/" and be relative to the named credential endpoint.
`serviceBinding`	string	Used to map non-supported media types for this external service registration to supported media types. Nillable. Available in API version 53.0 and later.
`serviceName`	string	The name of the cataloged API service that this external service registration belongs to. Available in API version 63.0 and later.
`status`	string	Required. Indicates service registration status. Valid values include: • `complete`—The API spec is valid and the registration is ready to use. • `incomplete`—The service registration hasn't completed.
`systemVersion`	int	The internal version of External Services used to register the API specification. Available in API version 55.0 and later. The system versions are independent of API versions. • `1`—Retired legacy External Services. • `2`—External Services with limitations on object and operation name length. • `3`—External Services automatically derives developer names fitting within 80 characters. • `4`—Removed the default character set when making a callout to an external service. If you want to specify a character set, include it in the OpenAPI spec, for example: `contentType: application/xml; charset=utf-8`. • `5`—Introduced asynchronous callouts with callbacks from Apex. • `6`—Current version. Added support for OAS discriminator mapping. This field is read-only. You can't upgrade an external service to a newer system version. If you want to take advantage of functionality in a newer system version, you must create an external service using the same OpenAPI spec and then replace any references to the old external service. See Register an External Service in Salesforce Help.

ExternalServiceOperation

Field Name	Field Type	Description
`active`	boolean	Required. Indicates whether the operation is active (`true`), or inactive (`false`).
`name`	string	Required. The operation's name.

Declarative Metadata Sample Definition

The following is an example of an ExternalServiceRegistration component that references an external credit service.

```
<?xml version="1.0" encoding="UTF-8"?>
<ExternalServiceRegistration xmlns="http://soap.sforce.com/2006/04/metadata">
  <label>BankService</label>
  <namedCredential>Bank</namedCredential>
  <schema>{
  "swagger": "2.0",
  "basePath": "/",
  "info": {
    "version": "1.0",
    "title": "External Service for demo bank",
    "description": "### External Service for demo bank",
    "x-vcap-service-name": "DemoBankRestServices"
  },
  ...
  "paths": {
  "/accounts/{accountName}": {
      ...
    }
  },
  "definitions": {
    "accountDetails": {
      ...
    },
    "errorModel": {
      ...
    }
  }
}</schema>
  <schemaType>OpenApi</schemaType>
  <schemaUrl>/accounts/schema</schemaUrl>
  <status>Complete</status>
</ExternalServiceRegistration>
```

serviceBinding

The following JSON-encoded string defines the mapping of a non-supported media type to a supported media type for external service request and response body serialization.

```
{"compatibleMediaTypes":{
    "application/x-acme-json":"application/json"
}}
```

The non-supported media type `application/x-acme-json` is mapped to the supported media type `application/json` for this External Services registration. The External Services runtime takes into account the non-supported media type for request and response header processing and serializes the request and response content by the mapped supported media type.

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SEE ALSO:

Salesforce Help: Media Type Mapping in External Service Registrations

ExtlClntAppConfigurablePolicies

Represents the policies for an external client app to disable or enable plugins.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

ExtlClntAppConfigurablePolicies components have the suffix `.ecaPlcy` and are stored in the `extlClntAppPolicies` folder.

Version

ExtlClntAppConfigurablePolicies components are available in API version 60.0 and later.

Special Access Rules

The View all External Client Apps, view their settings, and edit their policies user permission is required for users with admin roles to configure OAuth policies.

Fields

Field Name	Description
`externalClientApplication`	**Field Type** string **Description** Required. The name of the external client app associated with the plugins.
`isEnabled`	**Field Type** boolean **Description** Required. If `true`, all plugins are enabled unless individually disabled. If `false`, all plugins are disabled. The default value is `true`. Available in API version 60.0 and later.
`isMobilePluginEnabled`	**Field Type** boolean **Description** If `true`, the Mobile plugin is enabled. If `false` the Mobile plugin is disabled. The default value is `true`. Available in API version 63.0 and later.
`isNotificationPluginEnabled`	**Field Type** boolean **Description** If `true`, the Notification plugin is enabled. If `false` the Notification plugin is disabled. The default value is `true`. Available in API version 63.0 and later.
`isOauthPluginEnabled`	**Field Type** boolean **Description** If `true`, the OAuth plugin is enabled. If `false` the OAuth plugin is disabled. The default value is `true`. Available in API version 60.0 and later.
`isPushPluginEnabled`	**Field Type** boolean **Description** If `true`, the Push Notification plugin is enabled. If `false` the Push Notification plugin is disabled. The default value is `true`. Available in API version 63.0 and later.
`isSamlPluginEnabled`	**Field Type** boolean

Field Name	Description
	Description If `true`, the SAML plugin is enabled. If `false` the SAML plugin is disabled. The default value is `true`. Available in API version 63.0 and later.
`label`	**Field Type** string **Description** The OAuth policies name for the external client app.
`startPage`	**Field Type** ExtlClntAppStartPage (enumeration type of string) **Description** Determines which URL to use for the start page. • Custom • None • OAuth Available in API version 63.0 and later.
startUrl	**Field Type** string **Description** The custom URL where users are directed after they authenticate. For example, direct users to a specific page in the service provider app. Available in API version 63.0 and later.

Declarative Metadata Sample Definition

This example shows an ExtlClntAppConfigurablePolicies component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ExtlClntAppConfigurablePolicies xmlns="http://soap.sforce.com/2006/04/metadata">
    <externalClientApplication>myeca</externalClientApplication>
    <isEnabled>true</isEnabled>
    <isMobilePluginEnabled>true</isMobilePluginEnabled>
    <isNotificationPluginEnabled>true</isNotificationPluginEnabled>
    <isOauthPluginEnabled>true</isOauthPluginEnabled>
    <isPushPluginEnabled>true</isPushPluginEnabled>
    <isSamlPluginEnabled>true</isSamlPluginEnabled>
    <label>myecapolicy</label>
    <startPage>OAuth</startPage>
    <startUrl>https://example.org</startUrl>
</ExtlClntAppConfigurablePolicies>
```

This example `package.xml` references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>ExternalClientApplication</name>
    </types>
    <types>
        <members>*</members>
        <name>ExtlClntAppOauthSettings</name>
    </types>
    <types>
        <members>*</members>
        <name>ExtlClntAppGlobalOauthSettings</name>
    </types>
    <types>
        <members>*</members>
        <name>ExtlClntAppOauthConfigurablePolicies</name>
    </types>
    <types>
        <members>*</members>
        <name>ExtlClntAppConfigurablePolicies</name>
    </types>
    <version>60.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ExtlClntAppGlobalOauthSettings

Represents the global settings for the OAuth plugin in an external client app. These settings include private and sensitive OAuth consumer information that can't be packaged and must not be added to source control.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

ExtlClntAppGlobalOauthSettings components have the suffix `.ecaGlblOauth` and are stored in the `extlClntAppGlobalOauthSets` folder.

Version

ExtlClntAppGlobalOauthSettings components are available in API version 59.0 and later.

Special Access Rules

Access to the OAuth plugin requires orgs to enable the Allow Access to OAuth Consumer Secrets via Metadata API permission in Setup. The View External Client Apps Consumer Secrets in Metadata user permission is required for users with developer roles to configure global OAuth settings.

Fields

Field Name	Description
`callbackUrl`	**Field Type** string **Description** The endpoint that Salesforce calls back to your external client app during OAuth. It's the OAuth redirect_uri.
`certificate`	**Field Type** string **Description** If the app uses a certificate, the PEM-encoded certificate string. When provided, it enables the JWT Bearer flow. Available in API version 60.0 and later.
`consumerKey`	**Field Type** string **Description** A value used by the consumer for identification to Salesforce. Referred to as client_id in OAuth 2.0.
`consumerSecret`	**Field Type** string **Description** A value that is combined with the `consumerKey` and used by the consumer for identification to Salesforce. Referred to as client_secret in OAuth 2.0.
`externalClientApplication`	**Field Type** string **Description** Required. Name of the external client application.
`idTokenConfig`	**Field Type** ExternalAppIdTokenConfig **Description** The settings for the ID token.

Field Name	Description
`isClientCredentialsFlowEnabled`	**Field Type** boolean **Description** If set to `true`, the OAuth 2.0 client credentials flow is enabled. Available in API version 60.0 and later.
`isCodeCredFlowEnabled`	**Field Type** boolean **Description** If set to `true`, the external client app can use the Authorization Code and Credentials Flow and its variations for headless login, passwordless login, and guest user identity services in an off-platform app. Headless registration isn't currently supported for external client apps. The default value is `false`. To use this field, the Authorization Code and Credentials Flow must be enabled for your org in OAuth and OpenID Connect settings. Available in API version 61.0 and later.
`isCodeCredPostOnly`	**Field Type** boolean **Description** If set to `true`, for the Authorization Code and Credentials Flow, the external client app is required to send the user's credentials to the Salesforce `services/oauth2/authorize` endpoint in the body of a POST request. If set to `false`, the app can send a POST or GET request with the user's credentials in the request body or in a Basic authorization header. The default value is `false`. To use this field, the Authorization Code and Credentials Flow must be enabled for your external client app. Headless registration, a variation of this flow, isn't currently supported for external client apps. Available in API version 61.0 and later.
`isConsumerSecretOptional`	**Field Type** boolean **Description** If set to `false` (default), the external app's client secret is required in exchange for an access token in the OAuth 2.0 web server flow. If set to `true`, the external app's client secret is optional.
`isDeviceFlowEnabled`	**Field Type** boolean **Description** If set to `true`, the external client app can use the OAuth 2.0 device flow. Available in API version 60.0 and later.

Field Name	Description
`isIntrospectAllTokens`	**Field Type** boolean **Description** If set to `true`, authorizes the external app to introspect all access and refresh all tokens. If set to `false` (default), the external client app can introspect its own tokens.
`isNamedUserJwtEnabled`	**Field Type** boolean **Description** If set to `true`, the external client app issues JSON Web Token (JWT)-based access tokens. If set to `false`, it issues opaque access tokens. The default value is `false` Available in API version 61.0 and later.
`isPkceRequired`	**Field Type** boolean **Description** If set to `true` (default) Proof Key for Code for Exchange (PKCE) is required for OAuth integration. If set to `false`, PKCE is optional.
`isRefreshTokenRotationEnabled`	**Field Type** boolean **Description** If set to `true`, the refresh token rotation is enabled. Available in API version 60.0 and later.
`isSecretRequiredForRefreshToken`	**Field Type** boolean **Description** If set to `true` (default), the app's client secret is required in the authorization request of a refresh token and hybrid refresh token flow. If set to `false` and an app sends the client secret in the authorization request, Salesforce still validates it.
`isSecretRequiredForTokenExchange`	**Field Type** boolean **Description** If set to `true`, the app's client secret is required for token exchange. Available in API version 60.0 and later.
`isTokenExchangeEnabled`	**Field Type** boolean **Description** If set to `true`, token exchange is enabled. Available in API version 60.0 and later.

Field Name	Description
`label`	**Field Type** string **Description** External Client Application Global OAuth Settings name.
`shouldRotateConsumerKey`	**Field Type** boolean **Description** If set to `true`, the OAuth external client app's consumer key is replaced with a newly generated key on metadata deploy.. To maintain security, if this field is set to `true`, you must include the ignore warnings attribute in the deploy command. Default is `false`.
`shouldRotateConsumerSecret`	**Field Type** boolean **Description** If set to `true`, the OAuth external client app's consumer secret is replaced with a newly generated secret on metadata deploy. To maintain security, if this field is set to `true`, you must include the ignore warnings attribute in the deploy command. Default is `false`.

ExternalAppIdTokenConfig

Represents configurations that determine the ID token attributes.

Field Name	Description
`idTokenAudience`	**Field Type** string **Description** The audience that this ID token is intended for.
`idTokenIncludeAttributes`	**Field Type** boolean **Description** Indicates whether attributes are included in the ID token (`true`) or not (`false`).
`idTokenIncludeStandardClaims`	**Field Type** boolean **Description** Indicates whether standard claims about the authentication event are included in the ID token (`true`) or not (`false`).

Field Name	Description
idTokenValidityInMinutes	**Field Type** int **Description** The length of time that the ID token is valid for after it's issued. The value can be 1–720 minutes.

Declarative Metadata Sample Definition

This example shows an ExtlClntAppGlobalOauthSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ExtlClntAppGlobalOauthSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <callbackUrl>https://www.example.com</callbackUrl>
    <externalClientApplication>myeca</externalClientApplication>
    <idTokenConfig>
        <idTokenAudience>SalesforceAudience</idTokenAudience>
        <idTokenIncludeStandardClaims>true</idTokenIncludeStandardClaims>
        <idTokenValidityInMinutes>0</idTokenValidityInMinutes>
    </idTokenConfig>
    <isConsumerSecretOptional>false</isConsumerSecretOptional>
    <isIntrospectAllTokens>false</isIntrospectAllTokens>
    <isPkceRequired>true</isPkceRequired>
    <isSecretRequiredForRefreshToken>false</isSecretRequiredForRefreshToken>
    <label>myecaglobalset</label>
    <shouldRotateConsumerKey>false</shouldRotateConsumerKey>
    <shouldRotateConsumerSecret>false</shouldRotateConsumerSecret>
</ExtlClntAppGlobalOauthSettings>
```

This example `package.xml` references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>ExternalClientApplication</name>
    </types>
    <types>
        <members>*</members>
        <name>ExtlClntAppOauthSettings</name>
    </types>
    <types>
        <members>*</members>
        <name>ExtlClntAppGlobalOauthSettings</name>
    </types>
    <types>
        <members>*</members>
        <name>ExtlClntAppOauthConfigurablePolicies</name>
    </types>
    <types>
        <members>*</members>
```

```
        <name>ExtlClntAppConfigurablePolicies</name>
    </types>
    <version>60.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ExtlClntAppMobileConfigurablePolicies (Beta)

Represents an external client app's mobile policies configuration.

Note: The ExtlClntAppMobileConfigurablePolicies metadata type is a pilot or beta service that is subject to the Beta Services Terms at Agreements - Salesforce.com or a written Unified Pilot Agreement if executed by Customer, and applicable terms in the Product Terms Directory. Use of this pilot or beta service is at the Customer's sole discretion.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

ExtlClntAppMobileConfigurablePolicies components have the suffix `.ecaMobilePlcy` and are stored in the `extlClntAppMobilePolicies` folder.

Version

ExtlClntAppMobileConfigurablePolicies components are available in API version 64.0 and later.

Special Access Rules

There are no additional access requirements that are specific to this type.

Fields

Field Name	Field Type	Description
externalClientApplication	string	Required. The name of the associated external client app.
label	string	Label for the external client app's mobile policies configuration.
screenLockTimeout	ScreenLockTimeout (enumeration of type string)	When `isScreenLockEnabled` is true in the associated ExtlClntAppMobileSettings metadata type, `screenLockTimeout` represents the amount of time after which the mobile app locks and requires the app user to reauthenticate. Valid values include:

Field Name	Field Type	Description
		• `Never` • `One` (1 minute) • `Five` (5 minutes) • `Ten` (10 minutes) • `Thirty` (30 minutes) • `Sixty` (60 minutes) • `OneTwenty` (120 minutes) • `OneEighty` (180 minutes) • `TwoForty` (240 minutes)

ExtlClntAppMobileSettings (Beta)

Represents an external client app's mobile app settings, such as screen lock on a mobile device.

Note: The ExtlClntAppMobileSettings metadata type is a pilot or beta service that is subject to the Beta Services Terms at Agreements - Salesforce.com or a written Unified Pilot Agreement if executed by Customer, and applicable terms in the Product Terms Directory. Use of this pilot or beta service is at the Customer's sole discretion.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

ExtlClntAppMobileSettings components have the suffix `.ecaMobile` and are stored in the `extlClntAppMobileSettings` folder.

Version

ExtlClntAppMobileSettings components are available in API version 64.0 and later.

Special Access Rules

There are no additional access requirements that are specific to this type.

Fields

Field Name	Field Type	Description
externalClientApplication	string	Required. The name of the associated external client app.

Field Name	Field Type	Description
isScreenLockEnabled	boolean	Required. Indicates whether the mobile app locks the screen after a specified timeout value.
label	string	Label for the external client app's mobile app settings configuration.

ExtlClntAppNotificationSettings (Beta)

Represents an external client app's notification subscriptions for mobile.

Note: The ExtlClntAppNotificationSettings metadata type is a pilot or beta service that is subject to the Beta Services Terms at Agreements - Salesforce.com or a written Unified Pilot Agreement if executed by Customer, and applicable terms in the Product Terms Directory. Use of this pilot or beta service is at the Customer's sole discretion.

The ExtlClntAppNotificationSettings metadata type requires the OAuth plugin for External Client Apps. See OAuth Plugin Enablement with Metadata API in Salesforce Help.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

ExtlClntAppNotificationSettings components have the suffix `.ecaNotifications` and are stored in the `extlClntAppNotifSettings` folder.

Version

ExtlClntAppNotificationSettings components are available in API version 64.0 and later.

Special Access Rules

There are no additional access requirements that are specific to this type.

Fields

Field Name	Field Type	Description
externalClientApplication	string	Required. The name of the associated external client app.
label	string	Label for the external client app's notification settings configuration.
notificationTypes	ExtlClntAppNotificationType[]	A list of notification types the external client app is subscribed to. Only notifications of these types are returned to the associated external client app via API or sent as push notifications.

ExtlClntAppNotificationType

Represents a notification type that an external client app is subscribed to. Only custom notification types enabled for the mobile delivery channel are supported.

Note: You can use Notification Builder in Setup to configure a notification type for the mobile delivery channel. See Manage Notification Delivery Settings in Salesforce Help.

Field Name	Field Type	Description
notificationType	string	Required. The API name of the notification type.
pushByDefault	boolean	Required. Indicates whether the notification type is sent as a push notification on mobile devices. To send a notification type as a push notification, you must also configure the ExtlClntAppPushSettings metadata type.

ExtlClntAppOauthConfigurablePolicies

Represents the policies configured by the admin for an OAuth-enabled external client app.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

ExtlClntAppOauthConfigurablePolicies components have the suffix `.ecaOauthPlcy` and are stored in the `extlClntAppOauthPolicies` folder.

Version

ExtlClntAppOauthConfigurablePolicies components are available in API version 59.0 and later.

Special Access Rules

The View all External Client Apps, view their settings, and edit their policies user permission is required for users with admin roles to configure OAuth policies.

Fields

Field Name	Description
`apexHandler`	**Field Type** string

Field Name	Description
	Description Name of the Apex handler. Available in API version 61.0 and later.
`clientCredentialsFlowUser`	**Field Type** string **Description** The execution user for the OAuth 2.0 client credentials flow. Salesforce returns access tokens on behalf of this user. This user must have the API Only permission. Available in API version 60.0 and later.
`commaSeparatedCustomScopes`	**Field Type** string **Description** Custom scope names in a comma-separated list. Available in API version 61.0 and later.
`commaSeparatedPermissionSet`	**Field Type** string **Description** Permission set IDs in a comma-separated list. This field or commaSeparatedProfile is used when permittedUsersPolicyType is set to AdminApprovedPreAuthorized.
`commaSeparatedProfile`	**Field Type** string **Description** Profiles in a comma-separated list. This field or commaSeparatedPermissionSet is used when permittedUsersPolicyType is set to AdminApprovedPreAuthorized.
`customAttributes`	**Field Type** ExtlClntAppOauthPoliciesAttribute[] **Description** Unique attributes to be included as admin defaults. The maximum number accepted is 128. Each custom attribute must have a unique key and use an available field.
`executeHandlerAs`	**Field Type** string **Description** Username of the Apex handler's execution user. Available in API version 61.0 and later.
`externalClientApplication`	**Field Type** string **Description** Required.

Field Name	Description
	The name of the external client app associated with this OAuth policies file.
`guestJwtTimeout`	**Field Type** int **Description** The amount of time before a JWT-based access token issued to a guest user expires. Values are: • `1`—1 Minute • `5`—5 Minutes • `10`—10 Minutes • `15`—15 Minutes • `30`—30 Minutes Available in API version 61.0 and later.
`ipRelaxationPolicyType`	**Field Type** string **Description** The policy that determines IP restrictions. Values are: • `Enforce` • `Bypass` • `Bypass_2factor` • `Enforce_RelaxRefresh`
`isClientCredentialsFlowEnabled`	**Field Type** boolean **Description** If `true`, the client credentials flow is enabled. The default value is `false`. Available in API version 60.0 and later.
`isGuestCodeCredFlowEnabled`	**Field Type** boolean **Description** If `true`, the external client app can use the guest user variation of the Authorization Code and Credentials Flow. To use this flow variation, the external client app must also be configured to issue JWT-based access tokens. The default value is `false`. Available in API version 61.0 and later.
`isNamedUserJwtEnabled`	**Field Type** boolean

Field Name	Description
	Description If `true`, the external client app issues JWT-based access tokens instead of opaque access tokens. If this field is available, it means that the `isNamedUserJwtEnabled` field in the ExtlClntAppGlobalOauthSettings type is set to `true`. The default value is `false`. Deprecated.
`isTokenExchangeFlowEnabled`	**Field Type** boolean **Description** If `true`**true**, the token exchange flow is enabled. The default value is `false`. Available in API version 60.0 and later.
`label`	**Field Type** string **Description** The OAuth policies name for the external client app.
`namedUserJwtTimeout`	**Field Type** int **Description** The amount of time before a JWT-based access token issued to a named user expires. Values are: • `1`—1 Minute • `5`—5 Minutes • `10`—10 Minutes • `15`—15 Minutes • `30`—30 Minutes Available in API version 61.0 and later.
`permittedUsersPolicyType`	**Field Type** PermittedUsersPolicyType (enumeration of type string) **Description** The policy that determines which users are allowed in the external client app. Values are: • `AdminApprovedPreAuthorized` • `AllSelfAuthorized`
`policyAction`	**Field Type** PolicyAction (enumeration of type string)

Field Name	Description
	Description Requires users to verify their identity with two-factor authentication when they log in to the external client app. Use `RaiseSessionLevel` along with `requiredSessionLevel` to determine the security posture. Values are: • `Block` • `RaiseSessionLevel`
`refreshTokenPolicyType`	**Field Type** RefreshTokenPolicyType (enumeration of type string) **Description** The type of policy that determines when a token must be refreshed. Values are: • `Infinite` • `SpecificInactivity` • `SpecificLifetime` • `Zero`
`refreshTokenValidityPeriod`	**Field Type** int **Description** The number of units of measure used to specify validity when refresh token policy type is set to `SpecificInactivity` or `SpecificLifetime`.
`refreshTokenValidityUnit`	**Field Type** string **Description** The unit of measurement that is used to specify validity when refresh token policy type is set to `SpecificInactivity` or `SpecificLifetime`. Values are: • `Days` • `Hours` • `Months`
`requiredSessionLevel`	**Field Type** SessionSecurityLevel (enumeration of type string) **Description** Defines the security posture. Values are: • `HIGH_ASSURANCE`

Field Name	Description
	• `LOW` • `STANDARD`
`sessionTimeoutInMinutes`	**Field Type** int **Description** Length of time the external client app's session lasts.
`singleLogoutUrl`	**Field Type** string **Description** URL where Salesforce sends a logout request when users log out of Salesforce.
`startUrl`	**Field Type** string **Description** URL where users are directed after they authenticate.

ExtlClntAppOauthPoliciesAttribute

Represents admin-defined attributes that provide personal information to customize the external client app for a specific use case.

Field Name	Description
`formula`	**Field Type** string **Description** Required. The existing field that includes the desired information. For example, `Organization.Country`.
`key`	**Field Type** string **Description** Required. A unique name for the attribute. For example, `country`.

Declarative Metadata Sample Definition

This example shows an ExtlClntAppOauthConfigurablePolicies component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ExtlClntAppOauthConfigurablePolicies xmlns="http://soap.sforce.com/2006/04/metadata">
    <externalClientApplication>myeca</externalClientApplication>
    <label>myecapolicy</label>
    <apexHandler>MyEcaOauthApexHandler</apexHandler>
    <executeHandlerAs>admin@example.org</executeHandlerAs>
    <refreshTokenPolicyType>SpecificLifetime</refreshTokenPolicyType>
    <refreshTokenValidityPeriod>1</refreshTokenValidityPeriod>
    <refreshTokenValidityUnit>Days</refreshTokenValidityUnit>
    <ipRelaxationPolicyType>Enforce</ipRelaxationPolicyType>
    <permittedUsersPolicyType>AdminApprovedPreAuthorized</permittedUsersPolicyType>
    <commaSeparatedPermissionSet>PermSetExample</commaSeparatedPermissionSet>
    <commaSeparatedCustomScopes>CustomScopeExample</commaSeparatedCustomScopes>
    <sessionTimeoutInMinutes>1</sessionTimeoutInMinutes>
    <requiredSessionLevel>HIGH_ASSURANCE</requiredSessionLevel>
    <policyAction>RaiseSessionLevel</policyAction>
    <singleLogoutUrl>https://www.example.com</singleLogoutUrl>
    <startUrl>https://www.example.com</startUrl>
</ExtlClntAppOauthConfigurablePolicies>
```

This example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>ExternalClientApplication</name>
    </types>
    <types>
        <members>*</members>
        <name>ExtlClntAppOauthSettings</name>
    </types>
    <types>
        <members>*</members>
        <name>ExtlClntAppGlobalOauthSettings</name>
    </types>
    <types>
        <members>*</members>
        <name>ExtlClntAppOauthConfigurablePolicies</name>
    </types>
    <types>
        <members>*</members>
        <name>ExtlClntAppConfigurablePolicies</name>
    </types>
    <version>60.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ExtlClntAppOauthSettings

Represents the settings configuration for the external client app's OAuth plugin.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

ExtlClntAppOauthSettings components have the suffix `.ecaOauth` and are stored in the `extlClntAppOauthSettings` folder.

Version

ExtlClntAppOauthSettings components are available in API version 59.0 and later.

Special Access Rules

Access to the OAuth plugin requires orgs to enable the Allow Access to OAuth Consumer Secrets via Metadata API permission in Setup. The View External Client Apps Consumer Secrets in Metadata user permission is required for users with developer roles to configure OAuth settings.

Fields

Field Name	Description
`areAttributesIncludedInAssetToken`	**Field Type** boolean **Description** Indicates whether custom attributes associated with the external client app are included in the JSON Web Token (JWT) payload of an asset token issued as a result of the asset token flow. The default value is `false`. Available in API version 61.0 and later.
`areCustomPermsIncludedInAssetToken`	**Field Type** boolean **Description** Indicates whether custom permissions associated with the external client app are included in the JWT payload of an asset token issued as a result of the asset token flow. The default value is `false`.

Field Name	Description
	Available in API version 61.0 and later.
`assetTokenAudiences`	**Field Type** string **Description** Required for the OAuth asset token flow. The audience (`aud`) claim in the JWT payload of an asset token issued by the external client app. This claim identifies who the asset token is intended for. The value must be an array of case-sensitive strings, each containing a `StringOrURI` value. Specify an audience for each intended consumer of the asset token. Available in API version 61.0 and later.
`assetTokenSigningCertificate`	**Field Type** string **Description** Required for the asset token flow. The ID of the self-signed certificate used to sign asset tokens issued by the external client app. The certificate size is limited to 4 KB. If your certificate is too large, try using a DER-encoded file to reduce the size. Available in API version 61.0 and later.
`assetTokenValidity`	**Field Type** int **Description** Required for the asset token flow. The period of time for which the asset token is valid after it's issued, expressed as the number of seconds from 1970-01-01T0:0:0Z measured in UTC. The validity period must be within 3 minutes of the expiration time of the assertion. Available in API version 61.0 and later.
`clientAssertionCertificate`	**Field Type** string **Description** A certificate that's used to sign a client attestation JSON Web Token (JWT), which is required for requests to the OAuth 2.0 authorization challenge endpoint for headless identity flows for first-party apps. To confirm that the app that sent the request is your first-party app, Salesforce validates the client attestation JWT against this certificate.
`commaSeparatedOauthScopes`	**Field Type** string **Description** OAuth scopes for the external client app, written as a comma-separated list.

Field Name	Description
customAttributes	**Field Type** ExtlClntAppOauthSettingsAttribute[] **Description** Unique attributes to be included as developer defaults. The maximum number accepted is 128. Each custom attribute must have a unique key and use an available field.
externalClientApplication	**Field Type** string **Description** Required. The external client app associated with this OAuth plugin.
isFirstPartyAppEnabled	**Field Type** boolean **Description** Determines whether a first-party app can send requests to the OAuth 2.0 authorization challenge endpoint on this Experience Cloud site. This endpoint support headless identity flows using the OAuth 2.0 for First-Party Applications draft protocol.
label	**Field Type** string **Description** Label for the external client app.
oauthLink	**Field Type** string **Description** An auto-generated value that combines the org ID and the OAuth Consumer ID.
singleLogoutUrl	**Field Type** string **Description** URL where Salesforce sends a logout request when users log out of Salesforce.
trustedIpRanges	**Field Type** ExtlClntAppOauthIpRange[] **Description** Specifies the ranges of IP addresses that can access the app without requiring the user to authenticate with the external client app. The maximum number of IP ranges is 128.

ExtlClntAppOauthSettingsAttribute

Represents developer-defined attributes that are used to include additional information in the external client apps. Developers use these attributes to customize the app for specific use cases.

Field Name	Description
`formula`	**Field Type** string **Description** Required. The existing field that includes the desired information. For example, `Organization.Country`.
`key`	**Field Type** string **Description** Required. A unique name for the attribute. For example, `country`.

ExtlClntAppOauthIpRange

Represents the range of IP addresses that are trusted by the external client app.

Field Name	Description
`description`	**Field Type** string **Description** Identifies the purpose of the range, such as which part of a network corresponds to this range.
`endIpAddress`	**Field Type** string **Description** Required. Last address in the IP range, inclusive. Required with start address.
`startIpAddress`	**Field Type** string **Description** Required. First address in the IP range, inclusive. Required with end address.

Declarative Metadata Sample Definition

The following is an example of an ExtlClntAppOauthSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ExtlClntAppOauthSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <externalClientApplication>myeca</externalClientApplication>
    <label>My Oauth Settings</label>
    <trustedIpRanges>
        <startIpAddress>10.55.2.0</startIpAddress>
        <endIpAddress>10.55.2.255</endIpAddress>
        <description>Building 6</description>
    </trustedIpRanges>
    <trustedIpRanges>
        <startIpAddress>10.55.12.0</startIpAddress>
        <endIpAddress>10.55.12.255</endIpAddress>
    </trustedIpRanges>
    <customAttributes>
        <key>userattribute</key>
        <formula>User.Country</formula>
    </customAttributes>
    <commaSeparatedOauthScopes>Basic, Web</commaSeparatedOauthScopes>
</ExtlClntAppOauthSettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>ExternalClientApplication</name>
    </types>
    <types>
        <members>*</members>
        <name>ExtlClntAppOauthSettings</name>
    </types>
    <types>
        <members>*</members>
        <name>ExtlClntAppGlobalOauthSettings</name>
    </types>
    <types>
        <members>*</members>
        <name>ExtlClntAppOauthConfigurablePolicies</name>
    </types>
    <version>59.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ExtlClntAppPushConfigurablePolicies (Beta)

Represents an external client app's push notification policies configuration.

Note: The ExtlClntAppPushConfigurablePolicies metadata type is a pilot or beta service that is subject to the Beta Services Terms at Agreements - Salesforce.com or a written Unified Pilot Agreement if executed by Customer, and applicable terms in the Product Terms Directory. Use of this pilot or beta service is at the Customer's sole discretion.

The ExtlClntAppPushConfigurablePolicies metadata type requires the OAuth plugin for External Client Apps. See OAuth Plugin Enablement with Metadata API in Salesforce Help.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

ExtlClntAppPushConfigurablePolicies components have the suffix `.ecaPushPlcy` and are stored in the `extlClntAppPushPolicies` folder.

Version

ExtlClntAppPushConfigurablePolicies components are available in API version 64.0 and later.

Special Access Rules

There are no additional access requirements that are specific to this type.

Fields

Field Name	Field Type	Description
externalClientApplication	string	Required. The name of the associated external client app.
isFullContent	boolean	Required. Indicates if push notifications display the full notification title and body text (*`true`*). When set to *`false`*, standard notifications display a generic message and custom notifications display only the notification title.
label	string	Label for the external client app's push notification policies configuration.

ExtlClntAppPushSettings (Beta)

Represents an external client app's push notification settings.

Note: The ExtlClntAppPushSettings metadata type is a pilot or beta service that is subject to the Beta Services Terms at Agreements - Salesforce.com or a written Unified Pilot Agreement if executed by Customer, and applicable terms in the Product Terms Directory. Use of this pilot or beta service is at the Customer's sole discretion.

The ExtlClntAppPushSettings metadata type requires the OAuth plugin for External Client Apps. See OAuth Plugin Enablement with Metadata API in Salesforce Help.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

ExtlClntAppPushSettings components have the suffix `.ecaPush` and are stored in the `extlClntAppPushSettings` folder.

Version

ExtlClntAppPushSettings components are available in API version 64.0 and later.

Special Access Rules

There are no additional access requirements that are specific to this type.

Fields

Important: You can configure the ExtlClntAppPushSettings metadata type with either `androidPushConfig` or `applePushConfig`, but not both. You can configure `pushConfigLink` instead to refer to an existing `androidPushConfig` or `applePushConfig` record.

Of the three push notification configuration options (`androidPushConfig`, `applePushConfig`, or `pushConfigLink`), you must only have one of the options within the same record. If you create a record with `androidPushConfig` or `applePushConfig`, the `pushConfigLink` is automatically generated and retrievable from the metadata.

If you retrieve the ExtlClntAppPushSettings metadata of an existing packageable external client app to install in another org, delete the `androidPushConfig` or `applePushConfig` information from the record, if present. To make sure that the destination org refers to the information in the source org, keep the `pushConfigLink` field as the only push notification configuration in the record.

If the `pushServiceType` is *`Android`*, you must configure `androidPushConfig` or configure a `pushConfigLink` that refers to an existing `androidPushConfig` record. If the `pushServiceType` is *`Apple`*, you must configure `applePushConfig` or configure a `pushConfigLink` that refers to an existing `applePushConfig` record.

Field Name	Field Type	Description
androidPushConfig	ExtClntAppAndroidPushConfig	Represents the push notification configuration of an Android mobile app.
applePushConfig	ExtClntAppApplePushConfig	Represents the push notification configuration of an iOS mobile app.
externalClientApplication	string	Required. The name of the associated external client app.

Field Name	Field Type	Description
label	string	Label for the external client app's push notifications configuration.
pushConfigLink	string	Identifies the push notification credentials used by the app. Valid format is the org ID (for example, `00D000000000001`) and an `ExtlClntAppApplePushConfig` or `ExtlClntAppAndroidPushConfig` record ID (for example, `1Dh000000000001`) separated by a colon. For example: `00D000000000001:1Dh000000000001` If you configure `pushConfigLink`, you can't also have `androidPushConfig` or `applePushConfig` in the same record.
pushServiceType	PushServiceType (enumeration of type string)	Required. Identifies the mobile operating system of the mobile app. Valid values are: • `Apple` • `Android`

ExtlClntAppAndroidPushConfig

Represents the push notification configuration of an Android mobile app.

Field Name	Field Type	Description
fcmProject	string	Required. The ID of the Google Firebase project associated with the mobile app.
serviceAccount	string	Required. The Base64-encoded Admin SDK private key for your Google Firebase service account. You can generate this key from the Service accounts tab in the Google Firebase console. The maximum length of the string is 8000 characters.

ExtlClntAppApplePushConfig

Represents the push notification configuration of an iOS mobile app. To configure the required authentication for iOS push notifications, you submit either a private key (.p8 file) or a TLS certificate (.p12 file).

To configure push notifications with a private key (.p8 file), complete the `signingKey`, `keyIdentifier`, and `teamIdentifier` fields.

To configure push notifications with a TLS certificate (.p12 file), complete the `certificate` and `password` fields.

Field Name	Field Type	Description
applicationBundle	string	The bundle ID of the iOS mobile app from Apple App Store Connect.

Field Name	Field Type	Description
certificate	string	The Base64-encoded TLS certificate with Apple Push Notification service (APNs) enabled. To generate and export this certificate, see Communicate with APNs using a TLS certificate in Apple Developer documentation.
environment	ApplePushEnvironmentType (enumeration of type string)	Required. The Apple Push Notification service environment. Valid values are: • `Production` • `Sandbox`
keyIdentifier	string	The key identifier for the private key entered in the `signingKey` field. See Get a key identifier in Apple Developer documentation.
password	string	The password for the TLS certificate entered in the `certificate` field.
signingKey	string	The Base64-encoded private key with Apple Push Notification service (APNs) enabled. To generate and download this key, see Create a private key to access a service in Apple Developer documentation.
teamIdentifier	string	The team ID listed in the membership details of the Apple Developer account associated with the iOS mobile app.

ExtlClntAppSamlConfigurablePolicies

Represents SAML configuration policies for an external client app. Use this type to configure Salesforce as an identity provider for SAML single sign-on (SSO). In this type of SSO configuration, users log in to a third-party service provider, such as Google, using their Salesforce credentials.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

ExtlClntAppSamlConfigurablePolicies components have the suffix `.ecaSamlPlcy` and are stored in the `extlClntAppSamlConfigurablePolicies` folder.

Version

ExtlClntAppSamlConfigurablePolicies components are available in API version 63.0 and later.

Special Access Rules

To use the ExtlClntAppSamlConfigurablePolicies type, you must have the View all External Client Apps, view their settings, and edit their policies user permission.

This type must be related to a parent ExternalClientApplication. Because external client apps with SAML configurations can't be packaged, the `distributionState` for the parent external client app must be set to `Local`.

The parent external client app must also have an associated ExtlClntAppConfigurablePolicies metadata type where the `isSamlPluginEnabled` field is set to `true`.

Fields

Field Name	Description
`acsUrl`	**Field Type** string **Description** Required. The assertion consumer service (ACS) URL from the third-party service provider. The ACS URL is the endpoint where the service provider receives SAML responses from Salesforce.
`certificate`	**Field Type** string **Description** A security certificate that the third-party service provider uses to sign SAML requests. Include this field only if your service provider signs SAML requests and you also want to use a service provider-initiated SAML flow. If you include a certificate, Salesforce requires that all SAML requests from the service provider are signed. If you don't include a certificate, Salesforce accepts all SAML requests, whether or not they're signed.
`commaSeparatedPermissionSet`	**Field Type** string **Description** A comma-separated list of permission set IDs that defines the user permissions required for an end user to use the SAML SSO flow. The permission sets that you specify here apply to the entire app, not just its SAML configuration.
`commaSeparatedProfile`	**Field Type** string **Description** A comma-separated list of profile IDs that defines the profiles required for an end user to use the SAML SSO flow. Like permission sets, profiles define user permissions. The profiles that you specify here apply to the entire app, not just its SAML configuration. We recommend that you use permission sets to manage user permissions instead of profiles.
`customAttributes`	**Field Type** ExtlClntAppSamlConfigurablePoliciesAttribute[]

Field Name	Description
	Description Custom attributes that you can use to send more information about the user in SAML responses. For example, send information about the user's country. The service provider can use the information to validate the user's identity.
`encryptionCertificate`	**Field Type** string **Description** A certificate that's used to encrypt SAML assertions that Salesforce sends to the service provider. Use an X.509 certificate that's saved in your Certificate and Key Management settings. To get the certificate, work with a certificate provider. If you include an `encryptionCertificate`, make sure that your service provider is configured to decrypt SAML assertions.
`encryptionType`	**Field Type** ExtlClntAppSamlEncryptType **Description** If you include an `encryptionCertificate` to encrypt SAML assertions, the `encryptionType` field specifies the encryption method. When the service provider receives SAML assertions from Salesforce, it detects this method and decrypts it. These values are valid. • `AES_128`—Advanced Encryption Standard (AES) encryption algorithm with a 128-bit cryptographic key. • `AES_256`—AES encryption algorithm with a 256-bit cryptographic key. For more information about AES encryption from the National Institute of Standards and Technology, see https://www.nist.gov/publications/advanced-encryption-standard-aes-0.
`entityUrl`	**Field Type** string **Description** Required. The entity ID from the third-party service provider. The entity ID is a globally unique ID that Salesforce uses to recognize the service provider.
`externalClientApplication`	**Field Type** string **Description** Required. The `label` for the parent ExternalClientApplication.
`issuer`	**Field Type** string

Field Name	Description
	Description Specifies the URI from which Salesforce sends SAML responses. The service provider uses this value to confirm that the response came from Salesforce. If you don't include this field, Salesforce uses your My Domain by default. Include this field to specify a different value, such as an Experience Cloud site URL.
`label`	**Field Type** string **Description** A name for your external client app SAML policies configuration.
`nameIdFormat`	**Field Type** ExtlClntAppNameIdFormatType **Description** Specifies the format of the user's SSO identifier (dictated by the value of the `subjectType` field) in SAML messages. So that the service provider can recognize the user, the name ID format that Salesforce uses for SAML responses must match the format that the service provider uses. Get this value from your service provider. These values are valid. • `Unspecified` (default)—No format. • `EmailAddress`—The user's identifier is formatted as an email address. • `Persistent`—The user's identifier is in an opaque format. Only Salesforce and the service provider can recognize it. The identifier doesn't change based on context. • `Transient`—Like the `Persistent` identifier, the user's identifier is in an opaque format. But `Transient` identifiers are temporary values that can change.
`signingAlgorithmType`	**Field Type** ExtlClntAppSamlSignAlgoType **Description** The signing algorithm that Salesforce uses to secure SAML messages. The signing algorithm generates a signature by hashing the private key that's stored in the `certificate`. Salesforce includes this signature in the SAML response—in both the response body and in the SAML assertion. When the service provider receives SAML responses, it validates the signature. Salesforce also applies this algorithm to single logout requests and responses. These values are valid. • `SHA1`—Secure Hash Algorithm (SHA) 1 algorithm, which generates a 160-bit hash value. • `SHA256`—SHA-256 algorithm,which generates a 256-bit hash value.

Field Name	Description
`singleLogoutBindingType`	**Field Type** ExtlClntAppSamlBindingType **Description** The SAML HTTP binding type that the service provider uses when it initiates single logout. The binding type determines how the service provider transfers HTTP information to Salesforce. These values are valid. • `PostBinding`–The service provider uses POST requests for single logout. • `RedirectBinding`—The service provider sends single logout requests through the browser via GET requests.
`singleLogoutUrl`	**Field Type** string **Description** The SAML single logout endpoint on the service provider. When Salesforce initiates single logout, it sends logout requests to this endpoint.
`startUrl`	**Field Type** string **Description** A URL where users are directed after they authenticate. For example, direct users to a specific page in the service provider app.
`subjectCustomAttribute`	**Field Type** string **Description** If the `subjectType` is `CustomAttribute`, specify which field you want to use to identify the user. Choose an existing field on the User object, or create a custom User field.
`subjectType`	**Field Type** ExtlClntAppSamlSubjectType **Description** Specifies the user's SSO identifier. These values are valid. • `Username`—The user's Salesforce username. • `FederationId`—The user's federation ID, which maps to the `FederationIdentifier`—A field on the User object. The federation ID can be any value as long as both Salesforce and the service provider can recognize it. For example, get a value from the service provider and then specify it in Salesforce. • `UserId`—The user's 15-character Salesforce user ID. • `CustomAttribute`—An identifier that's taken from a custom field value. Specify the custom field in the `subjectCustomAttribute` field.

Field Name	Description
	• `PersistentId`—An opaque identifier that only Salesforce and the service provider recognize.

ExtlClntAppSamlConfigurablePoliciesAttribute

Represents custom attributes that provide more information about the user. The attributes are included in SAML assertions in SAML responses that Salesforce sends to the service provider.

Field Name	Description
`formula`	**Field Type** string **Description** Required. A field that stores the user information that you want to send. Format the value as `$<object name>.<field>`. For example: `$Organization.Country` to indicate the `Country` field on the Organization object.
`key`	**Field Type** string **Description** Required. A unique name for the attribute to help you and the service provider recognize it in SAML responses.

Declarative Metadata Sample Definition

The following is an example of an ExtlClntAppSamlConfigurablePolicies component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ExtlClntAppSamlConfigurablePolicies xmlns="http://soap.sforce.com/2006/04/metadata">
    <acsUrl>https://www.<serviceprovideracsurl>.com</acsUrl>
    <entityUrl>https://www.<serviceproviderentityid>.com</entityUrl>
    <externalClientApplication>mySamlEca</externalClientApplication>
    <issuer>https://mydomainname.my.salesforce.com</issuer>
    <label>myeca_samlpolicies</label>
    <nameIdFormat>Unspecified</nameIdFormat>
    <singleLogoutUrl>https://www.<serviceprovidersinglelogouturl>.com</singleLogoutUrl>
    <singleLogoutBindingType>RedirectBinding</singleLogoutBindingType>
    <subjectType>CustomAttribute</subjectType>
    <subjectCustomAttribute>MyCustomField</subjectCustomAttribute>
    <certificate>MIIDzDCCArQCCQCFaZKGsGqZ...</certificate>
    <encryptionCertificate>MIIDzDCCArQCCQCFaZKGsGqZ...</encryptionCertificate>
```

```
    <encryptionType>AES_128</encryptionType>
    <signingAlgorithmType>SHA1</signingAlgorithmType>
    <customAttributes>
        <key>User Firstname</key>
        <formula>$User.FirstName</formula>
    </customAttributes>
    <customAttributes>
        <key>User Country</key>
        <formula>$User.Country</formula>
    </customAttributes>
</ExtlClntAppSamlConfigurablePolicies>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>ExternalClientApplication</name>
    </types>
    <types>
        <members>*</members>
        <name>ExtlClntAppConfigurablePolicies</name>
    </types>
    <types>
        <members>*</members>
        <name>ExtlClntAppSamlConfigurablePolicies</name>
    </types>
    <version>63.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

FeatureParameterBoolean

Represents a boolean feature parameter in the Feature Management App (FMA). Feature parameters let you drive app behavior and track activation metrics in subscriber orgs that install your package. This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

FeatureParameterBoolean components have the suffix `.featureParameterBoolean`. The components are stored in the `featureParameters` folder, which contains components for all the feature parameter metadata types.

Version

FeatureParameterBoolean components are available in API version 41.0 and later.

Special Access Rules

Available to package developers who have access to the Feature Management App (FMA). For details, see Manage Features in the *Second-Generation Managed Packaging Developer Guide*.

Fields

Field Name	Field Type	Description
`dataFlowDirection`	FeatureParameterDataFlowDirection	After a package containing the components is installed, indicates whether the feature parameter's value is editable in your License Management Org (LMO) and read-only in your customer's org or the other way around.
`masterLabel`	string	The feature parameter name that appears in the user interface.
`value`	boolean	The default value for this feature parameter. You can reference this value in your code, just like you reference other values in a subscriber's org.

FeatureParameterDataFlowDirection

Represents the direction of the data flow between your License Management Org (LMO) and the customer's org.

Field Name	Field Type	Description
`FeatureParameterDataFlowDirection`	string	After a package containing the components is installed, indicates whether the feature parameter's value is editable in your License Management Org (LMO) and read-only in your customer's org or the other way around. Valid values are: • `LmoToSubscriber` • `SubscriberToLmo`

Declarative Metadata Sample Definition

The following is an example of a FeatureParameterBoolean component.

```
<?xml version="1.0" encoding="UTF-8"?>
<FeatureParameterBoolean xmlns="http://soap.sforce.com/2006/04/metadata">
    <dataflowDirection>SubscriberToLmo</dataflowDirection>
    <masterLabel>Budget Tracking Enabled</masterLabel>
```

```
    <value>false</value>
</FeatureParameterBoolean>
```

The following is an example `package.xml` that references the previous definition (and the definitions for the other feature parameter types).

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>FeatureParameterBoolean</name>
    </types>
    <types>
        <members>*</members>
        <name>FeatureParameterDate</name>
    </types>
    <types>
        <members>*</members>
        <name>FeatureParameterInteger</name>
    </types>
    <version>41.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

FeatureParameterDate

Represents a date feature parameter in the Feature Management App (FMA). Feature parameters let you drive app behavior and track activation metrics in subscriber orgs that install your package. This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

FeatureParameterDate components have the suffix `.featureParameterDate`. The components are stored in the `featureParameters` folder, which contains components for all the feature parameter metadata types.

Version

FeatureParameterDate components are available in API version 41.0 and later.

Special Access Rules

Available to package developers who have access to the Feature Management App (FMA). For details, see Manage Features in the *Second-Generation Managed Packaging Developer Guide*.

Fields

Field Name	Field Type	Description
`dataFlowDirection`	FeatureParameterDataFlowDirection	After a package containing the components is installed, indicates whether the feature parameter's value is editable in your License Management Org (LMO) and read-only in your customer's org or the other way around.
`masterLabel`	string	The feature parameter name that appears in the user interface.
`value`	date	The default value for this feature parameter. You can reference this value in your code, just like you reference other values in a subscriber's org.

FeatureParameterDataFlowDirection

Represents the direction of the data flow between your License Management Org (LMO) and the customer's org.

Field Name	Field Type	Description
`FeatureParameterDataFlowDirection`	string	After a package containing the components is installed, indicates whether the feature parameter's value is editable in your License Management Org (LMO) and read-only in your customer's org or the other way around. Valid values are: • `LmoToSubscriber` • `SubscriberToLmo`

Declarative Metadata Sample Definition

The following is an example of a FeatureParameterDate component.

```
<?xml version="1.0" encoding="UTF-8"?>
<FeatureParameterDate xmlns="http://soap.sforce.com/2006/04/metadata">
    <dataflowDirection>SubscriberToLmo</dataflowDirection>
    <masterLabel>Activation Date</masterLabel>
    <value>2017-10-23</value>
</FeatureParameterDate>
```

The following is an example `package.xml` that references the previous definition (and the definitions for the other feature parameter types).

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>FeatureParameterBoolean</name>
    </types>
    <types>
        <members>*</members>
        <name>FeatureParameterDate</name>
    </types>
    <types>
        <members>*</members>
        <name>FeatureParameterInteger</name>
    </types>
    <version>41.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

FeatureParameterInteger

Represents an integer feature parameter in the Feature Management App (FMA). Feature parameters let you drive app behavior and track activation metrics in subscriber orgs that install your package. This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

FeatureParameterInteger components have the suffix `.featureParameterInteger`. The components are stored in the `featureParameters` folder, which contains components for all the feature parameter metadata types.

Version

FeatureParameterInteger components are available in API version 41.0 and later.

Special Access Rules

Available to package developers who have access to the Feature Management App (FMA). For details, see Manage Features in the *Second-Generation Managed Packaging Developer Guide*.

Fields

Field Name	Field Type	Description
`dataFlowDirection`	FeatureParameterDataFlowDirection	After a package containing the components is installed, indicates whether the feature parameter's value is editable in your License Management Org (LMO) and read-only in your customer's org or the other way around.
`masterLabel`	string	The feature parameter name that appears in the user interface.
`value`	int	The default value for this feature parameter. You can reference this value in your code, just like you reference other values in a subscriber's org.

FeatureParameterDataFlowDirection

Represents the direction of the data flow between your License Management Org (LMO) and the customer's org.

Field Name	Field Type	Description
`FeatureParameterDataFlowDirection`	string	After a package containing the components is installed, indicates whether the feature parameter's value is editable in your License Management Org (LMO) and read-only in your customer's org or the other way around. Valid values are: • `LmoToSubscriber` • `SubscriberToLmo`

Declarative Metadata Sample Definition

The following is an example of a FeatureParameterInteger component.

```
<?xml version="1.0" encoding="UTF-8"?>
<FeatureParameterInteger xmlns="http://soap.sforce.com/2006/04/metadata">
    <dataflowDirection>SubscriberToLmo</dataflowDirection>
    <masterLabel>Current Project Count</masterLabel>
    <value>42</value>
</FeatureParameterInteger>
```

The following is an example `package.xml` that references the previous definition (and the definitions for the other feature parameter types).

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>FeatureParameterBoolean</name>
    </types>
```

```
    <types>
        <members>*</members>
        <name>FeatureParameterDate</name>
    </types>
    <types>
        <members>*</members>
        <name>FeatureParameterInteger</name>
    </types>
    <version>41.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

FieldRestrictionRule

Represents a field visibility rule that controls whether a field is visible to a user, based on the field's inclusion in a field set. If Enhanced Personal Information Management setting was enabled before Spring '22, field visibility is based on the field's compliance categorization. This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

FieldRestrictionRule components have the suffix `.rule` and are stored in the `fieldRestrictionRules` folder.

Version

FieldRestrictionRule components are available in API version 52.0 and later.

Special Access Rules

- To access this type, you must have the Manage Sharing permission.
- To create and manage Employee field visibility rules, you must be assigned a Workplace Command Center permission set license and the Provides access to Workplace Command Center features system permission.
- To create and manage User field visibility rules, you must enable Digital Experiences and the Enhanced Personal Information Management feature.

Fields

Field Name	Field Type	Description
`active`	boolean	Indicates whether the rule is active (`true`) or not (`false`). The default value is `false`.

Field Name	Field Type	Description
`classification`	string[]	Required. The data classification compliance categorization or field set that is targeted by the rule. The rule applies to fields that are marked with this categorization or included in this field set. If you enabled Enhanced Personal Information Management before Spring '22 (API version 54.0), you can use Salesforce's default compliance categorization values or values that you add yourself. If you enabled Enhanced Personal Information Management after Spring '22 (API version 54.0), use the PersonalInfo_EPIM field set or a field set that you add yourself.
`classificationType`	ClassificationType (enumeration of type string)	The type of classification method used in your org. If you enabled Enhanced Personal Information Management before Spring '22 (API version 54.0), use `ComplianceCategory`. If you enabled Enhanced Personal Information Management after Spring '22, use `FieldSet`. • `ComplianceCategory`— • `FieldSet`— The default value is `ComplianceCategory`. Available in API version 54.0 and later.
`description`	string	Required. The description of the rule.
`enforcementType`	EnforcementType (enumeration of type string)	Required. The type of rule. Possible values are: • `FieldRestrict`—Field visibility rule. Only this value is valid. • `Restrict`—Do not use. • `Scoping`—Do not use.
`masterLabel`	string	Required. The name of the rule.
`recordFilter`	string	Required. The criteria that determine which fields are visible to the specified users. For example, the field can check if the logged-in user matches the Employee's ID.
`targetEntity`	string	Required. The object for which you're creating the rule. Only the Employee and User objects are supported.
`userCriteria`	string	Required. The users that this rule applies to, such as all active users or users with a specified role or profile.
`version`	int	Required. The rule's version number.

Declarative Metadata Sample Definition

The following is an example of a FieldRestrictionRule component, which uses the ComplianceCategory classification type. The classification value is one of Salesforce's default compliance categorization values, but you can create a custom compliance categorization value to use instead.

```
<?xml version="1.0" encoding="UTF-8"?>
<FieldRestrictionRule xmlns="http://soap.sforce.com/2006/04/metadata">
    <active>true</active>
```

```
    <classification>PII</classification>
    <classificationType>ComplianceCategory</classificationType>
    <description>Is Owner of Employee</description>
    <enforcementType>FieldRestrict</enforcementType>
    <masterLabel>Is Owner Field Restriction Rule</masterLabel>
    <recordFilter>OwnerId = $User.Id</recordFilter>
    <targetEntity>Employee</targetEntity>
    <userCriteria>$User.IsActive = true</userCriteria>
    <version>1</version>
</FieldRestrictionRule>
```

The following is an example of a FieldRestrictionRule component, which uses the FieldSet classification type. The classification value is Salesforce's default field set for personal information, but you can create a field set to use instead.

```
<?xml version="1.0" encoding="UTF-8"?>
<FieldRestrictionRule xmlns="http://soap.sforce.com/2006/04/metadata">
    <active>true</active>
    <classification>PersonalInfo_EPIM</classification>
    <classificationType>FieldSet</classificationType>
    <description>Is Owner of Employee</description>
    <enforcementType>FieldRestrict</enforcementType>
    <masterLabel>Is Owner Field Restriction Rule</masterLabel>
    <recordFilter>OwnerId = $User.Id</recordFilter>
    <targetEntity>Employee</targetEntity>
    <userCriteria>$User.IsActive = true</userCriteria>
    <version>1</version>
</FieldRestrictionRule>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>FieldRestrictionRule</name>
    </types>
    <version>52.0</version>
</Package>
```

FlexiPage

Represents the metadata associated with a Lightning page. A Lightning page represents a customizable screen made up of regions containing Lightning components.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

This type extends the Metadata metadata type and inherits its `fullName` field.

Note: These pages are known as FlexiPages in the API, but are referred to as Lightning pages in the rest of the Salesforce documentation and UI.

Note: In API version 49.0 and later, arrays in a FlexiPage are represented as `valueList`. Each array element is represented as `valueListItem`, and the element name is represented as `value`. In API version 48.0 and earlier, arrays are represented as

`value` and array elements are formatted as a comma-separated list. Any FlexiPage retrieved using API version 49.0 or later uses `valueList` to represent component property array values, regardless of which API version was used to create the FlexiPage.

Lightning pages are used in several places.

- In the Salesforce mobile app, a Lightning page is the home page for an app that appears in the navigation menu.
- In Lightning Experience, Lightning pages can be used:
  - To customize the layout of record pages, the Salesforce Home page, and the Email Application pane in the Outlook and Gmail integrations.
  - As the home page for an app.
  - As the utility bar for a Lightning app.

For more information on Lightning pages, see Salesforce Help.

Note: The namespace prefix is important to help identify the source of items like fields, custom objects, and more. For example, when working with FlexiPages, we recommend keeping namespaces for object fields, because removing them can cause unexpected results such as name collisions.

File Suffix and Directory Location

FlexiPage components have the suffix `.flexipage` and are stored in the `flexipages` folder.

Version

FlexiPage components are available in API version 29.0 and later.

Fields

Field Name	Field Type	Description
`description`	string	The optional description text of the Lightning page.
events	FlexiPageEvent[]	The list of events associated with the Lightning page. This field is available in API version 53.0 and later.
`flexiPageRegions`	FlexiPageRegion[]	The list of regions of a page.
`masterLabel`	string	Required. The label for the Lightning page, which displays in Setup.
`pageTemplate`	string	Deprecated. Use this field in API versions 33.0 to 38.0 only. In later versions, use `template`. Required. The template associated with the Lightning page.
`parentFlexiPage`	string	The name of the Lightning page that this page inherits behavior from. This field is available in API version 37.0 or later.

Field Name	Field Type	Description
`platformActionlist`	PlatformActionList	The list of all actions, and their order, that display on a Lightning app page. In the Salesforce mobile app, the actions appear in the action bar. This field is available in API version 34.0 and later.
`quickActionList`	QuickActionList	The list of quick actions associated with the Lightning page.
`sobjectType`	string	The object the Lightning page is associated with. For Lightning pages of type `AppPage` or `HomePage`, this field is `null`. After the value of this field is set, it can't be changed. This field is available in API version 37.0 or later.
`template`	FlexiPageTemplateInstance	Required. The template associated with the Lightning page. This field is available in API version 39.0 and later.
`type`	FlexiPageType (enumeration of type string)	Required. The type of a page. In API versions 32.0 through 36.0, this field can only have a value of `AppPage`. Valid values are: • `CdpRecordPage`—A Lightning page that is used to override a CDPNearCoreObject record page in Lightning Experience. This value is available in API version 54.0 and later for orgs that have Data Cloud enabled. • `AppPage`—A Lightning page that is used as the home page for a custom app. • `CommAppPage`—A Lightning page that is used to represent a custom page, as created in the Experience Builder. This value is available in API version 37.0 and later. • `CommContractDetailViewPage`—This value is available in API version 64.0 and later. • `CommCheckoutPage`—A Lightning page that is used to create a B2B Commerce checkout, as created in the Experience Builder. This value is available in API version 46.0 and later. • `CommFlowPage` A Lightning page used to override a flow page, as created in the Experience Builder. This value is available in API version 45.0 and later. • `CommForgotPasswordPage`—A Lightning page that's used to override a forgot-password page, as created in Experience Builder. This value is available in API version 39.0 and later. • `CommFlowPage`—An out-of-the-box flow page, as created in Experience Builder. This value is available in API version 45.0 and later.

Field Name	Field Type	Description
		• `CommGlobalSearchResultPage` A Lightning page used to override the global search result page, as created in Experience Builder. This value is available in API version 41.0 and later. • `CommLoginPage`—A Lightning page that's used to override the login page, as created in Experience Builder. This value is available in API version 39.0 and later. • `CommNoSearchResultsPage`—An Experience Builder site page for B2B searches that return no results. The URL for this page is `no-results/:term`. The page starts out empty. You can add any component to it that accepts parameters to achieve the desired "no results" experience. For example, you can place an HTML Editor component or CMS components for recommendations, banners, help, and support. This value is available in API version 48.0 and later. • `CommObjectPage`—A Lightning page used to override an object page, as created in Experience Builder. This value is available in API version 38.0 and later. • `CommOrderComfirmationPage`—A Lightning page that is used to create a B2B Commerce order confirmation page in checkout, as created in the Experience Builder. This value is available in API version 46.0 and later. • `CommQuickActionCreatePage`—A Lightning page used to override the create record page, as created in Experience Builder. This value is available in API version 38.0 and later. • `CommRecordPage`—A Lightning page used to override a record page, as created in the Experience Builder. This value is available in API version 38.0 and later. • `CommRelatedListPage`—A Lightning page used to override a related list page, as created in the Experience Builder. This value is available in API version 38.0 and later. • `CommSearchResultPage`—A Lightning page used to override the search result page, as created in Experience Builder. This value is available in API version 38.0 and later. • `CommSelfRegisterPage`—A Lightning page used to override the self-registration page, as created in Experience Builder. This value is available in API version 39.0 and later. • `CommThemeLayoutPage`—A Lightning page used to override a theme layout page, as created in the Experience Builder. This value is available in API version 38.0 and later.

Field Name	Field Type	Description
		• `EmbeddedServicePage` This value is available in API version 45.0 and later. • `EmailContentPage` — A page that contains the builder markup for your email content. When you edit email content in the builder, the FlexiPage object remembers where you put the components. Because they include builder markup, you can't retrieve or deploy FlexiPages when type is EmailContentPage. • `EmailTemplatePage` — A page that contains the builder markup for your email template. When you edit an email template in the builder, the FlexiPage object remembers where you put the components. Because they include builder markup, you can't retrieve or deploy FlexiPages when type is EmailTemplatePage or EmailContentPage. • `ForecastingPage` —A Lightning page that is used to override the default forecasts page in Lightning Experience. This value is available in API version 57.0 and later. • `HomePage`—A Lightning page that is used to override the Home page in Lightning Experience. This value is available in API version 37.0 and later. • `MailAppAppPage`—An email application pane used to override the default layout in the Outlook and Gmail integrations. This value is available in API version 38.0 and later. • `OmniSupervisorPageType`–A Lightning page used to customize the user interface on the Omni-Channel Supervisor page. This value is available in API version 60.0 and later. • `RecordPage`—A Lightning page used to override an object record page in Lightning Experience. This value is available in API version 37.0 and later. • `RecordPreview` A Lightning page used to override standard lookup previews when hovering over previewable records in Lightning Experience.This value is available in API version 45.0 and later. • `UtilityBar`—A Lightning page used as the utility bar in Lightning Experience apps. This value is available in API version 38.0 and later. • `VoiceExtension`—A Lightning page used to customize user interfaces and agent actions in the

Field Name	Field Type	Description
		Omni-Channel widget for Service Cloud Voice. This value is available in API version 57.0 and later. This field is available in API version 32.0 and later.

FlexiPageEvent

An event associated with the Lightning page. Available in API version 53.0 and later.

Field Name	Field Type	Description
`sourceName`	string	Required. The name of the event source item. If the source is a custom Lightning web component, this field is the name of the component. In API 53.0, a source can be only a Lightning web component.
`sourceProperties`	FlexiPageEventSourceProperty[]	The list of properties associated with the event source.
`sourceType`	FlexipageEventSourceTypeEnum (enumeration of type string)	Required. The type of item assigned as the event source. In API version 53.0, this field can have only a value of `Component`.
`targets`	FlexiPageEventTarget[]	The list of targets associated with the event source.

FlexiPageEventSourceProperty

A property associated with an event. Available in API version 53.0 and later.

Field Name	Field Type	Description
`name`	string	Required. In API version 53.0 and later, the value of this field can be only `eventName`.
`value`	string	Required. If the `name` field value is `eventName`, this field is the name of the event. If the event source is a Lightning web component, this value must be the same as the event name defined in the source component's `js-meta.xml` file.

FlexiPageEventTarget

A target associated with an event source on the Lightning page. Available in API version 53.0 and later.

Field Name	Field Type	Description
`mappings`	FlexiPageEventPropertyMapping[]	A list of key-value pairs for an event's source-to-target bindings.
`method`	string	Required. The only valid value is `updateProperties`.
`name`	string	Required. The name of the event target. Valid values are: • `flexipage:componentService`
`properties`	FlexiPageEventTargetProperty[]	List of properties of the event target.
`type`	FlexiPageEventTargetTypeEnum (enumeration of type string)	Required. The type of item assigned as the event target. Valid values are: • `FlexipageServices`—A component on the Lightning page.

FlexiPageEventPropertyMapping

A key-value pair for an event's source-to-target bindings. Available in API version 53.0 and later.

Field Name	Field Type	Description
`name`	string	Required. Name of the target property that changes when the event is triggered.
`value`	string	Value of the target property when the event occurs. For properties of type string, integer, and boolean, you can use an expression to define their value. Valid expression format is `{!Event.eventPropertyName}`. Event is the only context supported for expressions in interactions.

FlexiPageEventTargetProperty

A property on the event source's target represented as a key-value pair. Available in API version 53.0 and later.

Field Name	Field Type	Description
`name`	string	Required. In API version 53.0 and later, the value of this field can be only `componentIdentifier`
`value`	string	Required. The ComponentInstance `identifier` value for the component.

FlexiPageRegion

FlexiPage Region represents the properties of a region of a page. A region can contain a record list component or a recent items component that can be scoped to a set of entities.

Note: A Lightning page region can contain up to 100 components.

Field Name	Field Type	Description
`appendable`	RegionFlagStatus (enumeration of type string)	This field is available in Digital Experiences in API 45.0 or later, but is reserved for future use for all other areas. Valid values are: • `disabled` • `enabled` This field is assessed in combination with `replaceable` and `prependable` • If all the properties are set to `enabled`, the region is unlocked • If all the properties are set to `disabled`, the region is locked • If none of the properties are specified OR any of these three properties are missing, the region is unlocked. This field is available in API version 35.0 or later.
`componentInstances`	ComponentInstance[]	Properties and name of the component instance. This field was removed in API version 49.0. In API version 49.0 and later, use the `itemInstances` field instead.
`itemInstances`	ItemInstance[]	Array of item instances, which can contain components and fields. This field is available in API version 49.0 or later.
`mode`	FlexiPageRegionMode (enumeration of type string)	This field is reserved for future use. Valid values are: • `Append` • `Prepend` • `Replace` This field is available in API version 35.0 or later.
`name`	string	Required. Unique name of the FlexiPage region.
`prependable`	RegionFlagStatus (enumeration of type string)	This field is available in Digital Experiences in API 45.0 or later, but is reserved for future use for all other areas. Valid values are: • `disabled`

Field Name	Field Type	Description
		• `enabled` This field is assessed in combination with `appendable` and `replaceable`. • If all the properties are set to `enabled`, the region is unlocked • If all the properties are set to `disabled`, the region is locked • If none of the properties are specified OR any of these three properties are missing, the region is unlocked. This field is available in API version 35.0 or later.
`replaceable`	RegionFlagStatus (enumeration of type string)	This field is available in Digital Experiences in API 45.0 or later, but is reserved for future use for all other areas. Valid values are: • `disabled` • `enabled` This field is assessed in combination with `appendable` and `prependable`. • If all the properties are set to `enabled`, the region is unlocked • If all the properties are set to `disabled`, the region is locked • If none of the properties are specified OR any of these three properties are missing, the region is unlocked. This field is available in API version 35.0 or later.
`type`	FlexiPageRegionType (enumeration of type string)	Required. The type of FlexiPage region. Valid values are: • `Background`—Represents a region for background utility items, which aren't visible in the UI. Supported for utility bars only. • `Facet` • `Region` This field is available in API version 35.0 or later.

ItemInstance

Instance of a component or field on a Lightning page. Available in API version 49.0 or later.

Field Name	Field Type	Description
`componentInstance`	ComponentInstance	Properties and name of the component instance.
`fieldInstance`	FieldInstance	API name, label, and visibility rule information of the field component. This field is available only on Lightning pages that use Dynamic Forms.

ComponentInstance

Instance of a component in a page, such as a filter list.

Field Name	Field Type	Description
`componentInstanceProperties`	ComponentInstanceProperty[]	The value of a single property in a component instance.
`componentName`	string	Required. The name of a single instance of a component.
`identifier`	string	Required. The unique name of the ComponentInstance. Provides a way to uniquely identify an individual instance of a component on a Lightning page. This field has a maximum limit of 120 characters. This field is available in API version 53.0 and later.
`visibilityRule`	UiFormulaRule	A set of one or more filters that define the conditions under which the component displays on the page. If the rule evaluates to `true`, the component displays on the page. If `false`, it doesn't display. If this field is `null`, the component displays by default. This field is available in API version 41.0 and later.

ComponentInstanceProperty

Value of a single property in a component instance. ComponentInstanceProperty has a maximum limit of 10,000 characters.

Field Name	Field Type	Description
`name`	string	Name of the property, unique within the component instance. For Lightning components, this value is the `<aura:attribute>` as defined in the `.cmp` file.
`type`	ComponentInstancePropertyTypeEnum (enumeration of type string)	If this field value is `null`, then the ComponentInstanceProperty values apply to the Lightning component. If this field value is `decorator`, then the ComponentInstanceProperty values apply to the *component decorator* for the Lightning component.

Field Name	Field Type	Description
		The component decorator is a wrapper around a Lightning component. The decorator can apply more capabilities to the component when it renders on a specific page in Lightning Experience. For example, you can configure a component decorator around a component on the Lightning Experience utility bar to set the component's height or width when opened. The `UtilityBar` is the only page type that supports component decorators. Valid values are: • `decorator` This field is available in API version 38.0 or later.
`value`	string	Reference or value of the property. When defining a Related List component, to use a parent record set the `parentFieldApiName` value to ***`object.field_name`***. If you don't want to use a parent record, set the value to ***`object.`***`Id`.
`valueList`	ComponentInstancePropertyList	An array of values in a component instance. Available in API version 49.0 and later.

Tabs

When you give a standard label to a tab in a Tabs component—such as Activity, Collaborate, or Details—and when the `name` field is set to `title`, the `value` field uses a system-defined value instead of the label. Here are some examples of the system-defined values:

- `Standard.Tab.activity`
- `Standard.Tab.collaborate`
- `Standard.Tab.detail`
- `Standard.Tab.feed`
- `Standard.Tab.preview`
- `Standard.Tab.relatedLists`

For example, let's say you have a Lightning page that contains a tab with the standard label "Activity". If you query the definition that page, you see the system-defined name of the tab, not the label, in `value`.

```
<componentInstances>
  <componentInstanceProperties>
    <name>title</name>
    <value>Standard.Tab.activity</value>
  </componentInstanceProperties>
    <componentName>flexipage:tab</componentName>
</componentInstances>
```

Save Options

Save options are available on pages of type `RecordPage` only, when users edit an account or when they create, edit, or clone a case or lead. Save options are configured as a ComponentInstanceProperty under FlexiPageTemplateInstance.

Set the ComponentInstanceProperty `name` to `saveOptions` and use `value` to define the checkbox values. The `value` field in this case is not a ComponentInstancePropertyList, but instead is a string representation of a JSON array of name and value pairs representing each checkbox name and its value.

API Name	Available Objects	Available Values	UI Label
UseDefaultAssignmentRule	Account	• `NONE` • `APPLY_OPTION_WITHOUT_CHECKBOX_DISPLAY` • `SHOW_CHECKBOX_WITH_DEFAULT_OFF` • `SHOW_CHECKBOX_WITH_DEFAULT_ON`	Evaluate this account against territory rules on save
UseDefaultAssignmentRule	Lead	• `NONE` • `SHOW_CHECKBOX_WITH_DEFAULT_OFF` • `SHOW_CHECKBOX_WITH_DEFAULT_ON`	Assign using active assignment rule
UseDefaultAssignmentRule	Case	• `NONE` • `APPLY_OPTION_WITHOUT_CHECKBOX_DISPLAY` • `SHOW_CHECKBOX_WITH_DEFAULT_OFF` • `SHOW_CHECKBOX_WITH_DEFAULT_ON`	Assign using active assignment rule
triggerOtherEmail	Case	• `NONE` • `SHOW_CHECKBOX_WITH_DEFAULT_OFF` • `SHOW_CHECKBOX_WITH_DEFAULT_ON`	Send notification email to Contact

Value	UI Result
`NONE`	Don't display the checkbox and don't apply any save options during save.
`APPLY_OPTION_WITHOUT_CHECKBOX_DISPLAY`	Don't display the checkbox, but apply the save option value during save.
`SHOW_CHECKBOX_WITH_DEFAULT_OFF`	Display the checkbox, unchecked by default.
`SHOW_CHECKBOX_WITH_DEFAULT_ON`	Display the checkbox, checked by default.

For example, you can set cases, when saved, to run the **Assign using active assignment rule** without displaying a checkbox, and display the **Send notification email to Contact** checkbox, checked by default.

```
saveOptions =
[{"name":"UseDefaultAssignmentRule","value":"APPLY_OPTION_WITHOUT_CHECKBOX_DISPLAY"},
{"name":"triggerOtherEmail","value":"SHOW_CHECKBOX_WITH_DEFAULT_ON"}]
```

Note: Set assignment rules, territory rules, and email templates before configuring them as save options.

ComponentInstancePropertyList

Value of an element in an array in a component instance.

Field Name	Field Type	Description
`valueListItems`	ComponentInstancePropertyListItem[]	An array of elements in a component instance.

ComponentInstancePropertyListItem

Name of an element in an array in a component instance.

Field Name	Field Type	Description
`value`	string	Name of an element in an array in a component instance.

In API version 49.0 and later, arrays in a FlexiPage are represented as `valueList`. Each array element is represented as `valueListItem`, and the element name is represented as `value`.

For example, if you have an array of actions with API names `Clone` and `Edit`, the array is represented as `valueList`, with two `valueListItems`. One `valueListItems` has the `value Clone`, and one `valueListItems` has the `value Edit`.

```
<componentInstances>
  <componentInstanceProperties>
    <name>actionApiName</name>
    <valueList>
      <valueListItems>
        <value>Clone</value>
      </valueListItems>
      <valueListItems>
        <value>Edit</value>
      </valueListItems>
    </valueList>
  </componentInstanceProperties>
</componentInstances>
```

UiFormulaRule

A set of one or more filters that define the conditions under which a component displays on a Lightning page. For example, you could construct a filter that causes a rich text component on an opportunity page to display only when the Amount is greater than $1,000,000. Available in API version 41.0 and later.

Field Name	Field Type	Description
`booleanFilter`	string	Specifies advanced filter conditions such as `1 AND 2`.
`criteria`	UiFormulaCriterion[]	List of one or more filters that, when evaluated, determine component visibility.

UiFormulaCriterion

A single filter that when evaluated, helps define component visibility on a Lightning page. Available in API version 41.0 and later.

Field Name	Field Type	Description
`leftValue`	string	Required. The field upon which the filter is based. For example, `AMOUNT`.
`operator`	string	Required. Defines the operator used to filter the data. Valid values are: • `CONTAINS` • `EQUAL` • `NE`—not equal • `GT`—greater than • `GE`—greater than or equal • `LE`—less than or equal • `LT`—less than
`rightValue`	string	The value by which you want to evaluate the component's visibility. For example, `1000000`.

You can use these expressions in the `leftValue` field when setting filters for component visibility.

- `{!$Client.FormFactor}`—Use this expression to control component visibility based on the device the page is being rendered on. Valid values are `Small` (phone), `Medium` (tablet), and `Large` (Lightning Experience desktop). Setting the value to `Small` for record pages is supported only in orgs that are enabled for the new Salesforce mobile app. This expression is supported for app pages in API version 41.0 and later, and record pages in API version 47.0 and later.
- `{!$Permission.CustomPermission.`***`permissionName`***`}`—Use this expression to control component visibility based on the custom permissions of the user viewing the Lightning page. Supported for app, Home, and record pages only.
- `{!$Permission.StandardPermission.`***`permissionName`***`}`—Use this expression to control component visibility based on the standard permissions of the user viewing the Lightning page. Supported for app, Home, and record pages only.
- `{!Record.`***`field`***`}`—Supported for record pages only.
- `{!$User.`***`field`***`}`—Supported for app, Home, and record pages only.

For example, to display a component only when it renders on a phone, add this filter: `{!$Client.FormFactor} EQUAL "SMALL"`. Or, to display a component only to the System Administrator, use `{!$User.Profile.Name} EQUAL "System Administrator"`.

Expressions in component visibility rules can span no more than five fields. For example, `{!Record.Account.Owner.Manager.Manager.Manager.LastName}` has six spans and therefore isn't supported.

FieldInstance

Represents a single field component that resides on a Lightning page. Available in API version 49.0 and later. This subtype is available only on Lightning Pages that have enabled Dynamic Forms.

Field Name	Field Type	Description
`fieldInstanceProperties`	FieldInstanceProperty on page 1104[]	Properties of the field instance. Contains a name and value pair for each property associated with the field.
`fieldItem`	string	The API name of the field, prefixed with its context. For example, record fields are prefixed with `Record.`.
`identifier`	string	Required. The unique name of the FieldInstance. Provides a way to uniquely identify an individual instance of a field on a Dynamic Forms-enabled Lightning page. This field has a maximum limit of 120 characters. This field is available in API version 53.0 and later.
`visibilityRule`	UiFormulaRule	A set of one or more filters that define the conditions under which the component displays on the page. If the rule evaluates to `true`, the component displays on the page. If `false`, it doesn't display. If this field is `null`, the component displays by default.

FieldInstanceProperty

Represents a single property of a field instance. Available in API version 49.0 and later. This subtype is available only on Lightning pages that have enabled Dynamic Forms.

Field Name	Field Type	Description
`name`	string	Name of the property, unique within the field instance. Valid values are: • `conditionalFormatRuleset` Available in API version 62.0 and later. • `uiBehavior` Available in API version 49.0 and later.
`value`	string	Reference or value of the property. When the `name` value is `uiBehavior`, valid values for this field are: • `None` • `Readonly` • `Required`

FlexiPageTemplateInstance

FlexiPageTemplateInstance represents an instance of a Lightning page template.

Field Name	Field Type	Description
`name`	string	Required. The name of a single instance of a template.
`properties`	ComponentInstanceProperty[]	The value of a single property in a template instance. Valid only for: • `CommThemeLayoutPage` • Dynamic Forms-enabled pages of type `RecordPage` that are associated with account, case, or lead objects Contains a name and value pair for each theme layout property associated with the page template. In Experience Builder, the theme layout and its properties appear in the Theme area.

PlatformActionList

PlatformActionList represents the list of actions, and their order, that display on a Lightning app page. Available in API version 34.0 and later.

Field Name	Field Type	Description
`actionListContext`	PlatformActionListContext (enumeration of type string)	Required. The context of the action list. Valid values are: • `Assistant` • `BannerPhoto` • `Chatter` • `Dockable` • `FeedElement` • `Flexipage` • `Global` • `ListView` • `ListViewDefinition` • `ListViewRecord` • `Lookup` • `MruList` • `MruRow` • `ObjectHomeChart` • `Photo` • `Record` • `RecordEdit` • `RelatedList` • `RelatedListRecord`

Field Name	Field Type	Description
`platformActionListItems`	PlatformActionListItem[]	The actions in the PlatformActionList.
`relatedSourceEntity`	string	When the `ActionListContext` is RelatedList or RelatedListRecord, this field represents the API name of the related list to which the action belongs.

PlatformActionListItem

PlatformActionListItem represents an action in the PlatformActionList. Available in API version 34.0 and later.

Field Name	Field Type	Description
`actionName`	string	Required. The API name for the action in the list.
`actionType`	PlatformActionType (enumeration of type string)	Required. The type of action. Valid values are: • `ActionLink`—An indicator on a feed element that targets an API, a web page, or a file, represented by a button in the Salesforce Chatter feed UI. • `CustomButton`—When clicked, opens a URL or a Visualforce page in a window or executes JavaScript. • `InvocableAction` • `ProductivityAction`—Productivity actions are predefined and attached to a limited set of objects. Productivity actions include Send Email, Call, Map, View Website, and Read News. Except for the Call action, you can't edit productivity actions. • `QuickAction`—A global or object-specific action. • `StandardButton`—A predefined Salesforce button such as New, Edit, and Delete.
`sortOrder`	int	Required. The placement of the action in the list.
`subtype`	string	The subtype of the action. For quick actions, the subtype is `QuickActionType`. For custom buttons, the subtype is `WebLinkTypeEnum`. For action links, subtypes are `Api`, `ApiAsync`, `Download`, and `Ui`. Standard buttons and productivity actions have no subtype.

Declarative Metadata Sample Definition

Here's a sample XML FlexiPage component definition for a custom opportunity record page. It includes a tab set and a rich text component with visibility rules assigned to it.

Note: As an Experience Builder site page, three initial regions in the definition show the `header` region as locked, the `content` region as unlocked, and the `footer` region as unlocked.

```
<?xml version="1.0" encoding="UTF-8"?>
<FlexiPage xmlns="http://soap.sforce.com/2006/04/metadata">
    <flexiPageRegions>
        <itemInstances>
            <componentInstance>
                <componentInstanceProperties>
                    <name>collapsed</name>
                    <value>false</value>
                </componentInstanceProperties>
                <componentInstanceProperties>
                    <name>hideChatterActions</name>
                    <value>false</value>
                </componentInstanceProperties>
                <componentInstanceProperties>
                    <name>numVisibleActions</name>
                    <value>3</value>
                </componentInstanceProperties>
                <componentName>force:highlightsPanel</componentName>
            </componentInstance>
        </itemInstances>
        <name>header</name>
        <type>Region</type>
    </flexiPageRegions>
    <flexiPageRegions>
        <itemInstances>
            <componentInstance>
                <componentInstanceProperties>
                    <name>hideUpdateButton</name>
                    <value>false</value>
                </componentInstanceProperties>
                <componentInstanceProperties>
                    <name>variant</name>
                    <value>linear</value>
                </componentInstanceProperties>
                <componentName>runtime_sales_pathassistant:pathAssistant</componentName>
            </componentInstance>
        </itemInstances>
        <name>subheader</name>
        <type>Region</type>
    </flexiPageRegions>
    <flexiPageRegions>
        <itemInstances>
            <componentInstance>
                <componentInstanceProperties>
                    <name>entityNames</name>
                    <valueList>
                        <valueListItems>
                            <value>Opportunity</value>
                        </valueListItems>
                    </valueList>
                </componentInstanceProperties>
```

```
            <componentInstanceProperties>
                <name>maxRecords</name>
                <value>3</value>
            </componentInstanceProperties>
            <componentName>flexipage:recentItems</componentName>
        </componentInstance>
    </itemInstances>
    <name>Facet-afbed70e-277a-41f5-9919-34651ff97773</name>
    <type>Facet</type>
</flexiPageRegions>
<flexiPageRegions>
    <itemInstances>
        <componentInstance>
            <componentInstanceProperties>
                <name>relatedListComponentOverride</name>
                <value>NONE</value>
            </componentInstanceProperties>
            <componentName>force:relatedListContainer</componentName>
        </componentInstance>
    </itemInstances>
    <name>facet-77f21b6f-ad73-4d79-838a-79e0df27cc63</name>
    <type>Facet</type>
</flexiPageRegions>
<flexiPageRegions>
    <itemInstances>
        <componentInstance>
            <componentName>force:detailPanel</componentName>
        </componentInstance>
    </itemInstances>
    <name>facet-c22fcfa7-d6f2-46ab-ac03-6c92e7398da1</name>
    <type>Facet</type>
</flexiPageRegions>
<flexiPageRegions>
    <itemInstances>
        <componentInstance>
            <componentName>runtime_sales_activities:activityPanel</componentName>
        </componentInstance>
    </itemInstances>
    <name>Facet-u9v2x6h8u4k</name>
    <type>Facet</type>
</flexiPageRegions>
<flexiPageRegions>
    <itemInstances>
        <componentInstance>
            <componentInstanceProperties>
                <name>body</name>
                <value>Facet-afbed70e-277a-41f5-9919-34651ff97773</value>
            </componentInstanceProperties>
            <componentInstanceProperties>
                <name>title</name>
                <value>Recent Items</value>
            </componentInstanceProperties>
            <componentName>flexipage:tab</componentName>
        </componentInstance>
```

```
        </itemInstances>
        <itemInstances>
            <componentInstance>
                <componentInstanceProperties>
                    <name>active</name>
                    <value>true</value>
                </componentInstanceProperties>
                <componentInstanceProperties>
                    <name>body</name>
                    <value>facet-77f21b6f-ad73-4d79-838a-79e0df27cc63</value>
                </componentInstanceProperties>
                <componentInstanceProperties>
                    <name>title</name>
                    <value>Standard.Tab.relatedLists</value>
                </componentInstanceProperties>
                <componentName>flexipage:tab</componentName>
            </componentInstance>
        </itemInstances>
        <itemInstances>
            <componentInstance>
                <componentInstanceProperties>
                    <name>body</name>
                    <value>facet-c22fcfa7-d6f2-46ab-ac03-6c92e7398da1</value>
                </componentInstanceProperties>
                <componentInstanceProperties>
                    <name>title</name>
                    <value>Standard.Tab.detail</value>
                </componentInstanceProperties>
                <componentName>flexipage:tab</componentName>
            </componentInstance>
        </itemInstances>
        <itemInstances>
            <componentInstance>
                <componentInstanceProperties>
                    <name>body</name>
                    <value>Facet-u9v2x6h8u4k</value>
                </componentInstanceProperties>
                <componentInstanceProperties>
                    <name>title</name>
                    <value>Standard.Tab.activity</value>
                </componentInstanceProperties>
                <componentName>flexipage:tab</componentName>
            </componentInstance>
        </itemInstances>
        <name>facet-27334405-c871-463f-bc20-b3713bbb4884</name>
        <type>Facet</type>
    </flexiPageRegions>
    <flexiPageRegions>
        <itemInstances>
            <componentInstance>
                <componentInstanceProperties>
                    <name>tabs</name>
                    <value>facet-27334405-c871-463f-bc20-b3713bbb4884</value>
                </componentInstanceProperties>
```

```
                <componentName>flexipage:tabset</componentName>
            </componentInstance>
        </itemInstances>
        <name>main</name>
        <type>Region</type>
    </flexiPageRegions>
    <flexiPageRegions>
        <itemInstances>
            <componentInstance>
                <componentInstanceProperties>
                    <name>decorate</name>
                    <value>true</value>
                </componentInstanceProperties>
                <componentInstanceProperties>
                    <name>richTextValue</name>
                    <value>&lt;p style=&quot;text-align: center;&quot;&gt;&lt;span
style=&quot;background-color: rgb(255, 255, 255); font-size: 18px; color: rgb(11, 11,
11);&quot;&gt;A million dollar opportunity closed! Oh yeah!&lt;/span&gt;&lt;/p&gt;</value>

                </componentInstanceProperties>
                <componentName>flexipage:richText</componentName>
                <visibilityRule>
                    <booleanFilter>1 AND 2</booleanFilter>
                    <criteria>
                        <leftValue>{!Record.Amount}</leftValue>
                        <operator>GE</operator>
                        <rightValue>1000000</rightValue>
                    </criteria>
                    <criteria>
                        <leftValue>{!Record.StageName}</leftValue>
                        <operator>EQUAL</operator>
                        <rightValue>Closed Won</rightValue>
                    </criteria>
                </visibilityRule>
            </componentInstance>
        </itemInstances>
        <itemInstances>
            <componentInstance>
                <componentInstanceProperties>
                    <name>decorate</name>
                    <value>true</value>
                </componentInstanceProperties>
                <componentInstanceProperties>
                    <name>richTextValue</name>
                    <value>&lt;p style=&quot;text-align: center;&quot;&gt;&lt;span
style=&quot;background-color: rgb(255, 255, 255); font-size: 16px; color: rgb(244, 0,
0);&quot;&gt;This component is for mobile users only.&lt;/span&gt;&lt;/p&gt;</value>
                </componentInstanceProperties>
                <componentName>flexipage:richText</componentName>
                <visibilityRule>
                    <criteria>
                        <leftValue>{!$Client.formFactor}</leftValue>
                        <operator>EQUAL</operator>
                        <rightValue>Small</rightValue>
```

```
                    </criteria>
                </visibilityRule>
            </componentInstance>
        </itemInstances>
        <itemInstances>
            <componentInstance>
                <componentName>forceChatter:recordFeedContainer</componentName>
            </componentInstance>
        </itemInstances>
        <name>sidebar</name>
        <type>Region</type>
    </flexiPageRegions>
    <masterLabel>New Opportunity Page</masterLabel>
    <sobjectType>Opportunity</sobjectType>
    <template>
        <name>flexipage:recordHomeWithSubheaderTemplateDesktop</name>
    </template>
    <type>RecordPage</type>
</FlexiPage>
```

And, here's the sample `package.xml` file that references the FlexiPage component definition:

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <fullName>New Opportunity Page</fullName>
    <types>
        <members>New_Opportunity_Page</members>
        <name>FlexiPage</name>
    </types>
    <version>49.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

Flow

Represents the metadata associated with a flow that encompasses the flow's structure, logic, and run-time behavior. It allows you to build dynamic applications that guide users through interactive screens, automate processes, and connect with various Salesforce and external services. This includes managing data operations like creating, updating, or deleting records, handling complex decisions, looping through collections, and invoking actions like Apex or external services to extend functionality. A flow contains options for API versioning, various execution environments, and detailed configuration of elements to design powerful automation solutions.

Important: Where possible, we changed non-inclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Here are the limitations related to how you can use Metadata API to work with flows:

- You can't use Metadata API to access a flow installed from a managed package unless the flow is a template.
- Spaces in a flow file name can lead to errors when you deploy the flow. You can include spaces at the beginning or end of a name, but these spaces are removed when you deploy the flow.

- You can deploy changes to an active flow if in a non-production org, such as a scratch or sandbox org. To deploy changes in a production org, you must enable the **Deploy processes and flows as active** preference. After you deploy changes to an active flow, the flow's detail page shows a new flow version that's active. The new version includes your changes.
- You can delete a flow version if it isn't active and doesn't have any paused interviews. If the flow version has paused interviews, wait for those interviews to resume and finish, or delete them.

Warning: Don't edit the metadata of retrieved Process Builder processes, such as flow components whose processType is `Workflow` or `InvocableProcess`. If you deploy process metadata that you edited, you can't open the process in the target org.

Declarative Metadata File Suffix and Directory Location

Flows are stored in the `Flow` directory of the corresponding package directory. The file name matches the flow's unique full name, and the extension is `.flow`.

Version

The flow Metadata API is available in API version 24.0 and later.

Flow

This metadata type represents a valid definition of a flow. This type extends the Metadata metadata type and inherits its `fullName` field.

Field Name	Field Type	Description
`actionCalls`	FlowActionCall[]	An array of nodes that defines calls to action. This field is available in API version 31.0 and later.
`apexPluginCalls`	FlowApexPluginCall[]	An array of nodes that defines calls to Apex plug-ins.
`apiVersion`	number	The API version that defines the execution behavior of the flow. This field is available in API version 50.0 and later. Flows created before API version 50.0 show an API version of 0 on the Flows list view in Setup. To show the correct API version number, create another version of the flow, and set the API version for running the flow to 49.0 or later.
`areMetricsLoggedToDataCloud`	boolean	Indicates whether the flow's metrics are logged to Data Cloud. The default value is `false`. This field is available in API version 63.0 and later.
`assignments`	FlowAssignment[]	An array of assignment nodes.
`choices`	FlowChoice[]	An array of static choice options.
`collectionProcessors`	FlowCollectionProcessor[]	An array of nodes that process collections. This field is available in API version 50.0 and later.
`constants`	FlowConstant[]	An array of constants.
`customErrors`	FlowCustomError[]	An array of custom errors.

Field Name	Field Type	Description
`customProperties`	FlowCustomProperty[]	An array of custom properties that specify flow properties such as the option to show a progress indicator in a screen flow. This field is available in API version 63.0 and later.
`decisions`	FlowDecision[]	An array of decision nodes.
`description`	string	Description of the flow.
`dynamicChoiceSets`	FlowDynamicChoiceSet[]	An array that constructs a set of choice options based on a database lookup.
`environments`	FlowEnvironment (enumeration of type string)	The environment in which the flow can run. Valid values are: • `Default`—The flow can run from a Visualforce component, Lightning page, flow action, or custom Aura component. • `Offline`—The flow can run only offline. Flow types that support offline flows must set this value. This value is available in API version 62.0 and later. • `Slack`—The flow can run in Slack and the default environment. You specify the Slack flow environment when you save the flow. This field is available in API version 55.0 and later.
`exitRules`	FlowExitRule[]	An array of exit rules that determine when to end the flow for a user in a segment-triggered flow. This field is available in API version 61.0 and later.
`experiments`	FlowExperiment[]	An array of experiments. This field is available in API version 61.0 and later.
`formulas`	FlowFormula[]	An array of formulas.
`fullName`	string	Required. Inherited from the Metadata component. Name of the file in Metadata API. A unique name for the flow that contains only underscores and alphanumeric characters. The name must be unique across the org, begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. To deploy or retrieve a version, you can specify the version number. For example, `sampleFlow-3` specifies version 3 of the flow whose unique name is sampleFlow. If you don't specify a version number, the flow is the latest version. In API version 43.0 and earlier, this field included the version number. In API version 44 and later, this field no longer includes the version number.

Field Name	Field Type	Description
interviewLabel	string	Label for the interview. This label helps users and administrators differentiate interviews from the same flow. In the user interface, this label appears in the Paused Flow Interviews component on the user's Home tab and in the list of paused flow interviews in Setup.
isAdditionalPermissionRequiredToRun	boolean	Override the default behavior and restrict access to enabled profiles or permission sets by setting this property to `true`. The default value is `false`. This field is available in API version 47.0 and later.
isTemplate	boolean	Indicates whether the process or flow is a template. The default value is `false`. When installed from managed packages, subscribers can't view or clone processes or flows because of intellectual property (IP) protection. But when those processes and flows are templates, subscribers can open them in a builder, clone them, and customize the clones. This field is available in API version 45.0 and later.
label	string	Required. Label for the flow.
loops	FlowLoop[]	An array of nodes for iterating through collections. This field is available in API version 30.0 and later.
migratedFromWorkflowRuleName	string	The name of the workflow rule that the flow was migrated from. This field is available in API version 54.0 and later.
orchestratedStages	FlowOrchestratedStage[]	An array of stage nodes in an orchestration. This field is available in API version 53.0 and later.
processMetadataValues	FlowMetadataValue[]	Metadata values for the flow. This field is available in API version 31.0 and later.
processType	FlowProcessType (enumeration of type string)	The type of the flow, as determined by the active version, or the latest version, if there's no active version. Valid values are: • `ActionableEventManagementFlow`—A flow that triggers an actionable event orchestration process in the background and automatically executes different types of actions based on the event type. This value is available in API version 62.0 and later. • `ActionCadenceAutolaunchedFlow`—A flow that's executed when a user completes a cadence step. This value is available in API version 56.0 and later. • `ActionCadenceStepFlow`—A screen flow used as a cadence step. This value is available in API version 56.0 and later. • `ActivityObjectMatchingFlow`—A flow that launches when Einstein Activity Capture detects and

Field Name	Field Type	Description
		captures a new activity, such as an email. This type of flow runs in the background without user interaction. This value is available with Sync Email as Salesforce Activity in API version 64.0 and later. • `Appointments`—A flow for Lightning Scheduler. This value is available in API version 44.0 and later. • `ApprovalWorkflow`—An orchestration that's used for an approval process. This value is available in API version 63.0 and later. • `AutoLaunchedFlow`—A flow that doesn't require user interaction. • `CheckoutFlow`—A flow used in Lightning B2B Commerce to create a checkout in a store. This value is available in API version 48.0 and later. • `ContactRequestFlow`—A flow that lets customers request to be contacted by customer support. This flow is used to create contact request records. This value is available in API version 45.0 and later. • `CustomerLifecycle`—A Salesforce Surveys flow that lets you associate survey questions with different stages in customer lifecycles. This value is available in API version 49.0 and later and only when the Customer Lifecycle Designer license is enabled. • `CustomEvent`—A process that is invoked when it receives a platform event message. In the UI, it's an event process. This value is available in API version 41.0 and later. • `DataCaptureFlow`— In the UI, Data Capture flows configure the Form tab in the Field Service mobile app. When the Data Capture flow is launched, its Flow metadata is publicly available in JavaScript format. This value is available in API version 62.0 and later. • `DcvrFrameworkDataCaptureFlow`—A screen flow that presents assessment questions from Discovery Framework. Launches when invoked by a user on a mobile device. This type of flow collects or displays information, requires user interaction, and works offline or online. This value is available in API version 62.0 and later. • `EvaluationFlow`—A flow for evaluating custom entry and exit conditions in an orchestration. Uses the `isOrchestrationConditionMet` output variable and discards values from any other output variables. This value is available in API version 54.0 and later. • `FieldServiceMobile`—A flow for the Field Service mobile app. This value is available in API version 39.0 and later.

Field Name	Field Type	Description
		• `FieldServiceWeb`—A flow for embedded Appointment Booking. Its UI label is Field Service Embedded Flow. This value is available in API version 41.0 and later. • `Flow`—A flow that requires user interaction because it contains one or more screens or local actions, choices, or dynamic choices. In the UI and Salesforce Help, it's a screen flow. Screen flows can be launched from the UI, such as with a flow action, Lightning page, or web tab. • `FSCLending`—A flow for Financial Services Cloud Mortgage. This value is available in API version 46.0 and later. • `IdentityUserRegistrationFlow`—A flow to handle user registration and updates for single sign-on with the authentication provider framework. Available in API version 64.0 and later. • `IndicatorResultFlow`—A flow for Outcome Management that calculates and creates indicator results for a selected indicator performance period. This value is available with the Outcome Management license in API version 60.0 and later. • `IndividualObjectLinkingFlow`—A flow that associates individuals with interactions such as voice calls, messaging sessions, or case-related emails. This value is available in API version 58.0 and later. • `InvocableProcess`—A process that another process or the Invocable Actions resource in REST API invokes. This value is available in API version 38.0 and later. • `Journey`—An audience-driven flow for Marketing Cloud. This value is available in API version 57.0 and later. • `LoginFlow`—A flow for login. This value is available in API version 51.0 and later. • `LoyaltyManagementFlow`—A flow for the Loyalty Management app that's invokable by loyalty program processes. This value is available in API version 54.0 and later. • `Orchestrator`—An orchestration that organizes flows into groups of steps contained in a series of stages. This value is available in API version 53.0 and later. • `PromptFlow`—A flow for Prompt Builder. Pass data between Prompt Builder and the flow. This value is available in API version 60.0 and later. • `RecommendationStrategy`—Build recommendations for your users. A recommendation launches its assigned flow. This value is available in API version 54.0 and later. See Flow Builder Strategies.

Field Name	Field Type	Description
		• `RoutingFlow`—A flow for Salesforce Omni-Channel routing and other business logic. This value is available in API version 52.0 and later. • `Survey`—A flow for Salesforce Surveys. From the UI, this type of flow is created in Survey Builder. This value is available in API version 42.0 and later. • `SurveyEnrich`—A Salesforce Surveys flow that uses the Survey Data Mapper. From the UI, this type of flow is created in the Survey Builder and requires an associated survey flow type. This value is available in API version 49.0 or later and only when the Customer Lifecycle Designer license is enabled. • `Workflow`—A process that is invoked when a record is created or edited. In the UI and Salesforce Help, it's a record change process. Across flow versions, you can change the type only from `Flow` to `AutoLaunchedFlow` or vice versa. Before you change the flow type, make sure that the flow contains only the elements, resources, and functionality that the new flow type supports. These values are reserved for future or Salesforce internal use. • `ActionCadenceFlow` • `ActionPlan` • `AppProcess` • `ApprovalWorkflow` • `CartAsyncFlow` • `DigitalForm` • `JourneyBuilderIntegration` • `LoginFlow` • `ManagedContentFlow` • `OrchestrationFlow` • `SalesEntryExperienceFlow` • `TransactionSecurityFlow` • `UserProvisioningFlow` This field is available in API version 31.0 and later.
`recordCreates`	FlowRecordCreate[]	An array of nodes for creating records in the database.
`recordDeletes`	FlowRecordDelete[]	An array of nodes for deleting records in the database.
`recordLookups`	FlowRecordLookup[]	An array of nodes for looking up records in the database.

Field Name	Field Type	Description
`recordRollbacks`	FlowRecordRollback[]	An array of nodes for rolling back transactions in the screen flow. This field is available in API version 52.0 and later.
`recordUpdates`	FlowRecordUpdate[]	An array of nodes for updating records in the database.
`runInMode`	FlowRunInMode (enumeration of type string)	The context that the flow runs in. Valid values are: • `DefaultMode`—How the flow is launched determines whether the flow runs in user context or in system context. In the UI, this value appears as **User or System Context—Depends on How Flow is Launched**. • `SystemModeWithSharing`—The flow respects org-wide default settings, role hierarchies, sharing rules, manual sharing, teams, and territories. The flow doesn't respect object permissions, field-level access, or other permissions of the running user. In the UI, this value appears as **System Context with Sharing—Enforces Record-Level Access**. • `SystemModeWithoutSharing`—The flow can access all data. In the UI, this value appears as **System Context without Sharing—Access All Data**. This value is available in API version 49.0 and later. This field is available in API version 48.0 and later.
`screens`	FlowScreen[]	An array of screen nodes.
`segment`	string	Reserved for future use.
`stages`	FlowStage[]	An array of stage resources that you can use throughout the flow. This field is available in API version 42.0 and later.
`start`	FlowStart[]	The flow's Start element, which specifies how and when the flow starts. This field is available in API version 47.0 and later.
`startElementReference`	string	Specifies which node or element is the starting point in the flow. This field isn't used in flows created or saved in Flow Builder in Winter '20 and later. Those flows use the `start` field instead to specify how the flow starts.
`status`	FlowVersionStatus (enumeration of type string)	The activation status of the flow. Valid values are: • `Active` • `Draft`—In the UI, this status appears as `Inactive`. • `Obsolete`—In the UI, this status appears as `Inactive`. • `InvalidDraft`—In the UI, this status appears as `Draft`.
`steps`	FlowStep[]	An array of step nodes.

Field Name	Field Type	Description
`subflows`	FlowSubflow[]	An array of subflows. This field is available in API version 25.0 and later.
`textTemplates`	FlowTextTemplate[]	An array of text templates.
`timeZoneSidKey`	string	The ID that defines the time zone in which the flow runs. This field is available in API version 56.0 and later.
`transforms`	FlowTransform[]	An array of data transformations. This field is available in API version 59.0 and later.
`triggerOrder`	int	The run order of a record-triggered flow, from 1 to 2,000. See Guidelines for Defining the Run Order of Record-Triggered Flows for an Object in Salesforce Help. This field is available in API version 54.0 and later.
`variables`	FlowVariable[]	An array of variable definitions.
`waits`	FlowWait[]	An array of wait nodes. This field is available in API version 32.0 and later.

FlowActionCall

Defines a call to an action from the flow. It extends FlowNode.

This metadata type is available in API version 31.0 and later.

Field Name	Field Type	Description
`actionCallPaths`	ActionCallPath[]	Reserved for future use.
`actionName`	string	Required. Name for the action. Must be unique across actions with the same `actionType`.
`actionType`	InvocableActionType (enumeration of type string)	Required. The action type. Valid values are: • `activateSessionPermSet`—Activates a session-based permission set for the running user. • `addMessageToChat`—Adds a message to an existing Salesforce Anywhere chat. This value is available in API version 49.0 and later. • `addMessageToQuipChat`—Adds a Quip message to an existing chat room. This value is available in API version 46.0 and later. • `addMessageToQuipDocument`—Adds a Quip message to an existing Quip document, spreadsheet, or slide. This value is available in API version 46.0 and later. • `addQuipDocumentToFolder`—Adds an existing Quip document, spreadsheet, or slide to an existing

Field Name	Field Type	Description
		folder. This value is available in API version 46.0 and later. • `addUsersToChat`—Adds users to an existing Salesforce Anywhere chat. This value is available in API version 49.0 and later. • `addUsersToQuipDocument`—Adds users, identified by their email addresses, to an existing Quip document, spreadsheet, or slide. This value is available in API version 46.0 and later. • `addUsersToQuipChat`—Adds users, identified by their email addresses, to an existing Quip chat room. This value is available in API version 46.0 and later. • `answerQuestionsWithSalesforceDocumentation`—Searches Salesforce documentation to provide answer to questions, as well as links to relevant articles. • `analyticsSendDigestAsSlackMsg`—Sends an Analytics digest to a Slack channel. This value is available in API version 64.0 and later. • `attachQuipDocumentToRecord`—Attaches a Quip document, spreadsheet, or slide to a Salesforce record. This value is available in API version 46.0 and later. • `apex`—Invokes an Apex method that has the @invocableMethod annotation. • `archiveKnowledgeArticles`—Archives a list of published Knowledge articles. This value is available in API version 45.0 and later. • `assignKnowledgeArticles`—Mass assigns knowledge articles from article list views. This value is available in API version 44.0 and later. • `automateRefund`—Initiate a refund to the customer. This value is available in API version 60.0 and later. • `buildIdentityVerification`—Calls an action that builds the identity verification context using the identity verification process definition specified in IdVerfProcessDefinition and information passed into the flow. Stores the result in the VerificationContext variable. This value is available in API version 55.0 and later. • `cdpPublishCalculatedInsight`—Run the calculated insight in Data Cloud. Available in API version 60.0 and later.

Field Name	Field Type	Description
		• `cdpPublishSegment`—Publish a segment in Data Cloud. Available in API version 60.0 and later. • `cdpRefreshDataStream`—Refresh a data stream in Data Cloud. Available in API version 60.0 and later. • `cdpRunIdentityResolution`—Runs a Data Cloud identity resolution process. This value is available in API version 57.0 and later. • `chat`—Creates a Salesforce Anywhere chat. This value is available in API version 49.0 and later. • `chatterPost`—Posts to Chatter. • `choosePricebook`—Selects a price book. • `contactRequestAction`—Creates a contact request record. This value is available in API version 45.0 and later. • `component`—Invokes the Lightning component that implements the `lightning:availableForFlowActions` interface and that is referenced by `actionName`. This value is available in API version 43.0 and later. • `contentWorkspaceEnableFolders`—Enables folders in a library. • `copyQuipDocument`—Creates a copy of an existing Quip document, spreadsheet, or slide, and gives it a new title. This value is available in API version 46.0 and later. • `createDraftFromOnlineKnowledgeArticle`—Creates a draft from a published knowledge article. This value is available in API version 45.0 and later. • `createFieldGnrnPromptTmplResp`—Creates a field generation prompt template response. This value is available in API version 62.0 and later. • `createInvoiceFromFulfillmentOrder`—Creates an invoice from a purchase order. Available to B2B Commerce. This value is available in API version 49.0 and later. • `createQuipChat`—Creates a Quip chat room. This value is available in API version 46.0 and later. • `createQuipDocument`—Creates a Quip document, spreadsheet, or slide. This value is available in API version 46.0 and later. • `createQuipFolder`—Creates a Quip folder. This value is available in API version 46.0 and later.

Field Name	Field Type	Description
		• `customNotificationAction`—Sends a custom notification. This value is available in API version 46.0 and later. • `dataCloudIngestionApi`—Send data to Data Cloud using Ingestion API. This value is available in API version 61.0 and later. • `deactivateSessionPermSet`—Deactivates a session-based permission set for the running user. • `deleteKnowledgeArticle`—Deletes a draft version (translation or master-language) or an entire archived knowledge article. This value is available in API version 46.0 and later. • `dynamicSendSurveyInvitation`—Sends customized notifications to users about important events or updates to the records that they're working on. This value is available in API version 51.0 and later. • `editQuipDocument`—Modifies the contents of an existing Quip document, spreadsheet, or slide. This value is available in API version 46.0 and later. • `einsteinDecidePath`—Determines a user's level of email engagement using Einstein Engagement Frequency or Einstein Engagment Scoring, and route users through the flow based on that engagement. This value is available in API version 64.0 and later. • `emailAlert`—Sends an email by referencing a workflow email alert • `emailSimple`—Sends an email by using flow resources • `exploreConversation`—Retrieves insights from a conversation. This value is available in API version 61.0 and later. • `externalConnector`—Executes a process or method exposed via a connector to an external system. This value is available in API version 63.0 and later. • `externalService`—Invokes an External Service operation that makes an HTTP request to an external system made available by an External Service schema registered through Setup. This value is available in API version 46.0 and later. • `findMatchingIndividuals`—Finds contact, lead, or employee records that match a search term. • `flow`—Invokes an autolaunched flow. This action type isn't available for flows with a processType of Flow or AutolaunchedFlow. To invoke an autolaunched flow

Field Name	Field Type	Description
		from one of those types, use FlowSubflow. This value is available in API version 32.0 and later. • `generateAiAgentResponse`—Generates a response from the AI agent based on input and instructions to support intelligent, conversational experiences. This value is available in API version 63.0 and later. • `generateAnalyticsAssetsContent`—Generates Analytics assets content. This value is available in API version 64.0 and later. • `generateVerificationCode`—Sends a verification code to the customer's email to verify their identity. This value is available in API version 63.0 and later. • `getActivitySummary`—Gets a summary of activity data associated with a specified record, including emails, calls, and meetings. This value is available in API version 60.0 and later. • `getArticleSmartLinkUrl`—Gets the Smart Link URL of the Salesforce Knowledge article. Smart links go to the right article and version, even when a new version is published or the URL name changes. This value is available in API version 54.0 and later. • `getResourcesForMnlScheduling`—Recommends resources to use to manually schedule the start of a care visit or recurring visits. You must enable Home Health to use this action. This value is available in API version 61.0 and later. • `getVerificationData`—Calls an invocable action to get verification data for selectedPrimaryVerificationContext and adds the results to selectedPrimaryVerificationContext. This value is available in API version 54.0 and later. • `goToCadenceStep`—Jumps to the specified step in the Sales cadence. This value is available in API version 57.0 and later. • `internalTestAction`—Reserved for internal use. • `internalTestConnectApiAction`—Reserved for internal use. • `limitRepetitions`—Limit the number of times the same recommendation or offer appears on the same record or for the same user during a time period in a recommendation strategy flow. This value is available in API version 55.0 and later.

Field Name	Field Type	Description
		• `lockRecord`—Lock or unlock a workflow-enabled or approval-enabled record for editing during an approval and specify who can edit the record while it's locked. • `massUpdateAccountForecast`—Bulk updates forecasts asynchronously. This value is available in API version 48.0 and later. • `massUpdateSalesAgreement`—Bulk updates sales agreements asynchronously. This value is available in API version 48.0 and later. • `processDataUsingGenAi`—Using Einstein generative AI, performs NLP to summarize text, extract key phrases, analyze sentiment, and unlock valuable insights. This value is available in API version 61.0 and later. • `publishActionableOrchSrcEvent`—Publishes events triggered by an external system. This value is available in API version 62.0 and later. • `publishKnowledgeArticles`—Mass publishes knowledge articles from article list views. This value is available in API version 44.0 and later. • `quickAction`—Invokes a QuickAction. • `rescheduleRecurringHomeVisits`—Reschedules all the home visits based on the recurrence pattern and scheduling policy provided. This value is available in API version 60.0 and later. • `restoreKnowledgeArticleVersion`—Restores an archived version of a knowledge article. This value is available in API version 45.0 and later. • `scheduleGroupVisits`—Create visiting records for patient home visits by bundling them into a group and scheduling either a single start-of-care visit or a series of recurring visits associated with the bundled records. This value is available in API version 60.0 and later. • `sendAlert`—Sends Salesforce Anywhere alerts to users. This value is available in API version 49.0 and later. • `sendNotification`—Sends an available notification type. This value is available in API version 54.0 and later. • `sendSurveyInvitation`—Sends email survey invitations to leads, contacts, and users in your org based on an action, such as when a customer support

Field Name	Field Type	Description
		case closes. This value is available in API version 47.0 and later. • `pardotSlackCompletionActionNotification`—Sends a user a Slack notification when a prospect completes an activity in Account Engagement. • `performSurveySentimentAnalysis`—Perform survey sentiment analysis to create or update the AI Sentiment Result records. This value is available in API version 55.0 and later. • `processDataUsingGenAi`—Using Einstein Generative AI, perform Natural Language Processing (NLP) to summarize text, extract key phrases, analyze sentiment, and unlock valuable insights. This value is available in API version 61.0 and later. • `skillsBasedRouting`—Creates a PendingServiceRouting record used for Omni-Channel skills-based routing. This value is available in version 44.0 and later. • `slackArchiveChannel`—Archives a Slack channel in a Slack workspace. This value is available in API version 54.0 and later. • `slackCheckUsersAreConnectedToSlack`—Indicates whether a collection of Salesforce users is connected to a given Slack app. This value is available in API version 54.0 and later. • `slackCreateChannel`—Creates a Slack channel in a Slack workspace. This value is available in API version 54.0 and later. • `slackGetConversationInfo`—Retrieves the name of a Slack channel or group direct message and finds out whether it's archived. This value is available in API version 54.0 and later. • `slackInviteUsersToChannel`—Adds users who are connected to a given Slack app to a Slack channel or group direct message. This value is available in API version 54.0 and later. • `slackPinMessage`—Pin or unpin a message in a Slack channel or group direct message. This value is available in API version 54.0 and later. • `slackPostMessage`—Send a message to a Slack channel or group direct message. This value is available in API version 54.0 and later. • `slackSendMessageToLaunchFlow`—Send a message to a Slack channel, direct message, or the

Field Name	Field Type	Description
		Messages tab of a Slack app that includes a button that a recipient can use to launch a screen flow. This value is available in API version 55.0 and later. • `slackUpdateMessage`—Edits a message that was previously sent to a Slack channel or group direct message. This value is available in API version 54.0 and later. • `submitKnowledgeArticleForTranslation`—Submits a published or draft knowledge article for translation. This value is available in API version 46.0 and later. • `submit`—Submits a record for approval. • `transformNlpActionResult`—Transform an AI Natural Language Processing (NLP) result created by using Einstein Generative AI into an Apex Object record. This value is available in API version 61.0 and later. • `verifyCustomerCode`—Verifies the code entered by the customer to complete identity verification. This value is available in API version 63.0 and later. These values are used in Omnichannel Inventory. If no version is specified, the value is available in API version 51.0 and later. • `ociCreateReservation`—Creates one or more inventory reservations at a location or location group. • `ociFulfillReservation`—Fulfills one or more inventory reservations at a location. • `ociGetAvailability`—Gets inventory availability data for one or more products at one or more inventory locations or location groups. • `ociReleaseReservation`—Releases one or more inventory reservations. • `ociTransferReservation`—Transfers one or more inventory reservations between locations or location groups. These values are used in the B2B Commerce Checkout Flow. If no version is specified, the value is available in API version 47.0 and later. • `updateCheckoutSessionStateAction`—Updates the checkout session next state for checkout flows. This value is available in API version 49.0 and later. • `priceCart`—Requests prices for all items in a cart during B2B Commerce checkout. This value is available in API version 47.0 and later.

Field Name	Field Type	Description
		• `checkoutSessionAction`—Initiates or retrieves an existing Checkout Session for Checkout Flows. Available to B2B Commerce. This value is available in API version 49.0 and later. • `cancelCartAsyncOperation`—Cancels a WebCart's async operation. Available to B2B Commerce. This value is available in API version 49.0 and later. • `calcCartPromotionsAction`—Requests a full cart promotion calculation of all applicable line items in the Web Cart during B2B Commerce checkout. This value is available in API version 52.0 and later. • `checkCartInventoryAction`—Requests an inventory for all items in a Web Cart during B2B Commerce checkout. This value is available in API version 47.0 and later. • `calcCartShipmentAction`—Calculates the shipping cost for all items in a Web Cart during B2B Commerce checkout. This value is available in API version 47.0 and later. • `cartToOrderAction`—Creates a Salesforce Standard Order in draft mode. This value is available in API version 47.0 and later. • `activateOrderAction`—Activates a draft order, which creates an order summary. This value is available in API version 47.0 and later. These values are used in the B2B Commerce and D2C Commerce. • `recordTaxReversal`—Reverses the recorded tax transactions in an external system. This value is available in API version 62.0 and later. • `recordTaxTransaction`—Records tax transactions from an order summary to an external system. This value is available in API version 62.0 and later. For values used in Business Rules Engine, see Flow for Business Rules Engine. These values are used in the Commerce Checkout Flow. If no version is specified, the value is available in API version 55.0 and later. • `addCartItem`—Adds an item to a cart during Commerce checkout. • `createCart`—Creates a cart during Commerce checkout.

Field Name	Field Type	Description
		• `deleteCart`—Deletes a cart during Commerce checkout. These values are used in Salesforce CMS Workflows and Approvals. If no version is specified, the value is available in API version 58.0 and later. • `managedContentPublishVariant`—Publishes a content variant associated with a flow. This value is available in API version 59.0 and later. • `managedContentRoleStepInteractive`—Assigns a content variant review to a CMS role. • `managedContentUnpublishVariant`—Unpublishes a published content variant associated with a flow. This value is available in API version 59.0 and later. • `managedContentVariantSetLockStatus`—Sets the locked status of a content variant. • `managedContentVariantSetReadyStatus`—Sets the ready for publication status of a content variant. For values used in Digital Lending, see Flow for Digital Lending. These values are used in Order Management. If no version is specified, the value is available in API version 48.0 and later. • `addOrderItemSummarySubmit`—Adds order item summaries to an order summary. This value is available in API version 54.0 and later. • `adjustOrderItemSummariesPreview`—Previews the expected results of applying a price adjustment to order item summaries from an order summary without actually applying it. This value is available in API version 49.0 and later. • `adjustOrderItemSummariesSubmit`—Applies a price adjustment to order item summaries from an order summary. This value is available in API version 49.0 and later. • `authorizePayment`—Authorizes a card payment. This value is available in API version 55.0 and later. • `cancelFulfillmentOrderItem`—Removes items from a fulfillment order. • `cancelOrderItemSummariesPreview`—Previews the expected results of canceling order item summaries from an order summary without actually canceling them.

Field Name	Field Type	Description
		• `cancelOrderItemSummariesSubmit`—Cancels order item summaries from an order summary. • `confirmHeldFulfillmentOrderCapacity`—Confirms held fulfillment order capacity. This value is available in API version 55.0 and later. • `createCreditMemoOrderSummary`—Creates a credit memo for an order summary. • `createFulfillmentOrder`—Creates one or more fulfillment orders and fulfillment order products for an order delivery group summary, which defines a recipient and delivery method. • `createFulfillmentOrders`—Creates fulfillment orders and fulfillment order products for multiple order delivery group summaries, each of which defines a recipient and delivery method. This value is available in API version 51.0 and later. • `createInvoiceFromChangeOrders`—Creates an invoice for one or more change orders. This value is available in API version 56.0 and later. • `createInvoiceFromFulfillmentOrder`—Creates an invoice for a fulfillment order. • `createOrderPaymentSummary`—Creates an order payment summary for an authorization or payments belonging to an order summary. • `createOrderSummary`—Creates an order summary for an order. • `createReturnOrder`—Creates a return order and return order items for an order. • `ensureFundsOrderSummaryAsync`—Triggers an asynchronous background process to ensure funds through a payment provider for an invoice belonging to an order summary. • `ensureRefundsOrderSummaryAsync`—Triggers an asynchronous background process to ensure refunds through a payment provider for an invoice belonging to an order summary. • `getFulfillmentOrderCapacityValues`—Gets fulfillment order capacity information. This value is available in API version 55.0 and later. • `holdFulfillmentOrderCapacity`—Holds fulfillment order capacity. This value is available in API version 55.0 and later. • `orderRoutingFindRoutesWithFewestSplits`—Evaluates ordered product quantities against available inventory

Field Name	Field Type	Description
		to determine the smallest combination of locations that can fulfill the order. This value is available in API version 51.0 and later. • `orderRoutingFindRoutesWithFewestSplitsUsingOCI`—Evaluates ordered product quantities against available inventory at specified location groups and locations to determine the smallest combination of locations that can fulfill the order. This value is available in API version 54.0 and later. • `orderRoutingRankByAverageDistance`—Calculates the average distance from sets of inventory locations to an order recipient, and returns the sets sorted by that average distance. This value is available in API version 51.0 and later. • `releaseHeldFulfillmentOrderCapacity`—Releases held fulfillment order capacity. This value is available in API version 55.0 and later. • `returnOrderItemSummariesPreview`—Previews the expected results of returning order item summaries from an order summary without actually returning them. • `returnOrderItemSummariesSubmit`—Returns order item summaries from an order summary. • `returnReturnOrderItems`—Processes return order line items. These values are used in the Employee Service. If no version is specified, the value is available in API version 63.0 and later. • `createServiceRequestCase`—Creates a case or incident for the requested service. • `getLeaveBalance`—Gets the leave balance of a specific employee. For values used in Financial Services Cloud, see Flow for Financial Services Cloud. For values used in Fundraising for Nonprofit Cloud, see Flow for Fundraising. For values used in Health Cloud, see Flow for Health Cloud. For values used in Insurance Brokerage, see Flow for Insurance Brokerage. For values used in Insurance Policy Administration, see Flow for Insurance Policy Administration. For values used in Life Sciences Cloud, see Flow for Life Sciences Cloud.

Field Name	Field Type	Description
		For values used in Manufacturing Cloud, see Flow for Manufacturing Cloud. These values are used in Rebate Management. • `addRebateMemberList`—Adds a list of members to a rebate program. This value is available in API version 51.0 and later. • `calculateProjectedRebateAmount`—Calculates the projected rebate amount for rebate types associated with a specified transaction ID. This value is available in API version 54.0 and later. • `calculateRebateAmountAndUpsertPayout`—Calculates the rebate amount and upserts the rebate payout for the specified aggregate record. This value is available in API version 51.0 and later. • `getBenefitAndCalculateRebateAmount`—Gets benefit details, and optionally calculates the rebate amount for the specified aggregate record. This value is available in API version 51.0 and later. • `getEligibleProgramRebateTypes`—Retrieves the eligible program rebate types for a mapped object. This value is available in API version 52.0 and later. • `generateRebatePayoutPeriods`—Generates payout periods for a rebate program based on the frequency specified in the program. This value is available in API version 51.0 and later. • `processRebatesBatchCalculationJob`—Processes a rebate batch calculation job from the Data Processing Engine. This value is available in API version 51.0 and later. • `processProgramRebateTypeProducts`—Insert or delete records in the Program Rebate Type Product object. This value is available in API version 53.0 and later. • `rebatesProcessCSV`—Processes an uploaded CSV file using Bulk API 2.0 and converts the file's data into records in the target object. This value is available in API version 51.0 and later. • `upsertCustomRebatePayout`—Upserts the custom calculated rebate payout for the specified aggregate record. This value is available in API version 51.0 and later. For values used in Referral Marketing, see Flow for Referral Marketing. These values are used in Loyalty Management.

Field Name	Field Type	Description
		• `adjustPoints`—Adjusts loyalty points for a specified program member or journal transaction. This value is available in API version 51.0 and later. • `assignTierBenefits`— Assigns Member Benefits to a member tier for benefits that are associated with a Benefit Action. This value is available in API version 51.0 and later. • `cancelAccrual`—Cancels a specific set of accrual transactions. • `creditPoints`—Credits loyalty points to a specified program member's balance. This value is available in API version 51.0 and later. • `cancelRedemption`—Reverts a specific set of redemption transactions. This value is available in API version 51.0 and later. • `changeTier`—Changes the tier for a specified program member. This value is available in API version 51.0 and later. • `changeTierWhenNoErrors`—Changes tier for a specified loyalty program member only when all the input parameters meet the criteria. This value is available in API version 51.0 and later. • `debitPoints`—Debits loyalty points to a specified program member's balance. This value is available in API version 51.0 and later. • `executeMemberBenefit`—Processes the benefit action associated with the benefit, which is assigned to a loyalty program member. This value is available in API version 51.0 and later. • `generateMemberReferralCode`—Generates a unique 8-character referral code for a loyalty program member. This value is available in API version 57.0 and later. • `getMemberActiveSegments`—Retrieve active Data Cloud market segments that a loyalty program member is a part of. • `getTier`—Gets the current tier for a specified program member. This value is available in API version 51.0 and later. • `getPointsBalance`—Gets the loyalty points balance for a specified program member. This value is available in API version 51.0 and later.

Field Name	Field Type	Description
		• `getLoyaltyPromotion`—Gets active loyalty promotions based on a transaction journal. This value is available in API version 53.0 and later. • `getLoyaltyPromotionBasedOnSalesforceCDP`—Gets promotions for a member based on the market segment the member belongs to. This value is available in API version 53.0 and later. • `issueVoucher`—Issues a voucher for a member or contract. This value is available in API version 51.0 and later. • `mergeLoyaltyProgramMembership`—Merges two active loyalty program member records that both belong to the same loyalty program. This value is available in API version 56.0 and later. • `transferMemberPointsToGroups`—Transfers points from an individual member or a corporate member to the member's associated group. This value is available in API version 53.0 and later. • `updateProgressForCumulativePromotionUsage`—Updates the progress a member has made towards attaining a cumulative type promotion. This value is available in API version 53.0 and later. • `unmergeLoyaltyProgramMembership`—Unmerges loyalty program member records that have a Merged status. The action unmerges memberships in the Merged status from the previously merged membership. This value is available in API version 56.0 and later. • `runProgramProcess`—Triggers an active loyalty program process. This value is available in API version 56.0 and later. • `runProgramProcessForTransactionJournal`—Triggers an active loyalty program process whose process type is TransactionJournal. This value is available in API version 54.0 and later. These values are for Decision Table. • `decisionTableAction`—Runs an active decision table definition. This value is available in API version 51.0 and later. • `refreshDecisionTable`—Refreshes the decision table cache. This value is available in API version 51.0 and later. These values are for the Batch Management jobs.

Field Name	Field Type	Description
		• `batchJobAction`—Runs the batch management jobs definitions. This value is available in API version 51.0 and later. • `submitFailedRecordsBatchJob`—Resubmits an existing batch job with failed records for processing. This value is available in API version 52.0 and later. This value is for Data Processing Engine. • `dataProcessingEngineAction`—Runs the data processing engine definitions. This value is available in API version 51.0 and later. This value is used for Einstein Visit Recommendation. • `saveRecommendationDecision`—Save visit and task recommendation decisions. This value is available in API version 51.0 and later. These values are used in flows for Engagement. • `createEngagementsDetailsRep`—Creates a JSON representation. Use the details of Engagement Interaction, Messaging Session, and Voice Call records; the related Engagement Topic and Note records; and transcripts to create the representation. • `getConversationTranscripts`—Gets the list of transcripts of the conversations between an agent and a customer. • `getEngagements`—Gets engagement interaction, messaging session, and voice call records associated with a specified account record. • `getRecordDetails`—Gets the details of specified records, including the name of the parent record. These values are used in Field Service. If no version is specified, the value is available in API version 52.0 and later. • `addWorkPlans`—Creates work plan and work step objects from the work plan library. • `addWorkSteps`—Creates work step objects from the work plan library. • `deleteWorkPlans`—Deletes all the work plans and work steps associated with a work order or work order line item. • `generateWorkPlans`—Generates work plans based off rules defined in the work plan library. For values used in Intelligent Form Reader, see Flow for Intelligent Form Reader.

Field Name	Field Type	Description
		For values used in Intelligent Document Reader, see Flow for Intelligent Document Reader. This value is used in Public Sector Solutions. • `createBenefitDisbursement`—Creates a benefit disbursement for an eligible benefit assignment. This value is available in API version 57.0 and later. • `runRecordAggrBatchProcDef`—Runs a Data Processing Engine definition to process an asynchronous batch job that creates or updates record aggregation results. This value is available in API version 59.0 and later. For values used in Net Zero Cloud, see Flow for Net Zero Cloud. For values used in Asset Lifecycle, see Flow for Asset Lifecycle. For values used in Salesforce Pricing, see Flow for Salesforce Pricing. For values used in Dynamic Revenue Orchestrator, see Flow for Dynamic Revenue Orchestrator. For values used in Product Discovery, see Flow for Product Discovery. For values used in Rate Management, see Flow for Rate Management. For values used in Quote and Order Capture, see Flow for Quote and Order Capture. For values used in Billing, see Flow for Billing. For values used in Usage Management, see Flow for Usage Management. This value is used in Einstein generative AI features. • `generatePromptResponse`—Generates a response based on the large language model (LLM) response for the specified prompt template and inputs. This value is available in API version 60.0 and later. These values are used in Einstein Bots. If no version is specified, the value is available in API version 56.0 and later. • `getDataCategoryDetails`—Gets the labels and API names for a specified data category associated with the Knowledge object and its child categories. • `getDataCategoryGroups`—Gets the labels and API names of the active data category groups associated with the Knowledge object that's visible to the current user.

Field Name	Field Type	Description
		• `searchKnowledgeArticles`—Searches for knowledge articles with specified search terms, language, data category group, and data category. These values are reserved for future use. • `thanks` • `metricRefresh` • `exportSurveyResponses` • `rpa` This value is used for Einstein Initiate Language Processing Action. • `initiateNaturalLangProcessing`—Create a record for the AI natural language processing result and initiate text processing by using the service specified in the related record. This value is available in API version 60.0 and later. These value are used in Einstein Work summaries. • `getCaseInfoToSummarize`—Gets case field details like subject, description, comments, emails, and conversation transcripts for use with prompt templates in Prompt Studio. This value is available in API version 63.0 and later. • `getConvTrscpForRecord`—Gets the conversation transcript associated with a VoiceCall, MessagingSession, or LiveChatTranscript record. This value is available in API version 63.0 and later. This value is used in Media Cloud. • `vlocity_cmt__MediaIntegrationProcedureInvocable`—Ca an Integration Procedure from a Salesforce Flow to process media content. This value is available in API version 60.0 and later. These values are used in the Get Forecast Guidance flow. • `getForecastContext`— Gets the forecast context for a specified user. This value is available in API version 61.0 and later. • `getForecastOpportunities` — Gets forecast opportunities for a user that matches the specified criteria. This value is available in API version 61.0 and later. This value is used in the Get Opportunity Details flow. • `getRecPrioData`— Gets the record data and field metadata required to prioritize records. This value is available in API version 62.0 and later.

Field Name	Field Type	Description
		This value is used in Einstein Case Classification flow. • `applyCaseClassificationRecommendations`—takes a Case ID as input and outputs a case SObject with recommendations applied. This value is available in API version 57.0 and later. These values are used in the Activities: Match Email to Records flow. Sync Email as Salesforce Activity must be enabled. • `associateRecordsWithActivity`—Updates the specified email message to associate it with specified account, contact, lead, opportunity, and user records. This value is available in API version 63.0 and later. • `getAcctOpptyFromEmailAddr`—Gets an account record associated with one of the specified contacts or unmatched email addresses and also gets an opportunity record related to the account. This value is available in API version 63.0 and later. • `getContcLeadsFromEmailAddr`—Matches email addresses to contact and lead records related to specified active user records. This value is available in API version 63.0 and later. • `getUsersFromEmailAddresses`—Gets user records with email addresses that match those specified in the To, From, or CC address field after a sent email is captured by Einstein Activity Capture. This value is available in API version 63.0 and later. These values are used in the Identity User Registration flow. • `generateUserData`—Generates placeholder user data for the fields that are required to create a user. Available in API version 64.0 and later. • `getUserDataFromJsonString`—Gets an attribute value from a JSON object that has been serialized into a string. Use this action to retrieve user information from the identity provider's ID token and user info response. Available in API version 64.0 and later. These values are used in the Contracts flow. • `checkInContractDocumentVersion`—Check-in a contract document version. Available in API version 64.0 and later. • `checkOutContractDocVersion`—Check-out a contract document version. Available in API version 64.0 and later.

Field Name	Field Type	Description
		• `createClmContract`—Create a contract for a specified record. Available in API version 64.0 and later. • `createContractDocumentVersion`—Create a contract version for a specified record. Available in API version 64.0 and later. • `getCntntDocDtlForCntrDocVer`—Get content document details for the contract document version. Available in API version 64.0 and later. • `getContractDocumentVersions`—Get contract document versions. Available in API version 64.0 and later. • `performContractAction`—Perform actions on a contract based on its status. Available in API version 64.0 and later. • `sendContractForESignature`—Send a contract to specified recipients for e-signature. Available in API version 64.0 and later. • `unlockContractDocumentVersion`—Unlock an active contract document version that the user previously locked. Available in API version 64.0 and later. • `updateClmContracts`—Update contract for a specified record and update or create associated contract documents. Available in API version 64.0 and later. These values are used in the Einstein GPT Usecases flow. • `createCaseForFinclAcctAddrUpdt`—Create a case to update the financial account address. Available in API version 64.0 and later. • `createVisitForContextRecord`—Create a visit to record context. Available in API version 64.0 and later. • `draftAGiftProposal`—Create a gift proposal for an account. Available in API version 64.0 and later. • `getCardDetailsForAccount`—Get card details for an account. Available in API version 64.0 and later. • `getFinancialAccountAddresses`—Get financial account address details for an account. Available in API version 64.0 and later. • `getFinancialTransactions`—Get all financial account transactions associated with a specific financial account. Available in API version 64.0 and later.

Field Name	Field Type	Description
		• `summarizeMedicalHistoryForPatient`—Summarize medical history of a specified patient. Available in API version 64.0 and later. • `summarizeMedicationDetailsForPatient`—Summarize medication details of a specified patient. Available in API version 64.0 and later.
`connector`	FlowConnector	Specifies which node to execute after this action call.
`dataTypeMappings`	FlowDataTypeMapping[]	An array of data type mappings for input and output values that have the generic sObject data type. This field is available in API version 48.0 and later.
`einsteinDecidePath`	string	Reserved for future use.
`faultConnector`	FlowConnector	Specifies which node to execute if the action call results in an error.
`flowTransactionModel`	FlowTransactionModel (enumeration of type string)	Required. Specifies the transactional model for flows that execute invocable actions. Valid values are: • `Automatic`— Creates a transaction if the invocable action supports it and there's pending DML. • `CurrentTransaction`— Keeps the invocable action running in the same transaction. • `NewTransaction`— Creates a transaction before the invocable action is executed. This field is available in API version 51.0 and later.
`inputParameters`	FlowActionCallInputParameter[]	An array of input parameters from the flow to the action.
`isWaitUntilCompleted`	boolean	Specifies whether to pause the flow until the action is completed. This field is available in API version 61.0 and later.
`nameSegment`	string	Specifies the name of the versioned action. Supported only when nameSegment is specified. This field is available in API version 58.0 to 61.0. This field is deprecated in API version 62.0 and later.
`offset`	int	Specify the number of months, days, hours, or minutes to pause the flow while it waits for the action to be completed. This field is available in API version 61.0 and later.
`offsetUnit`	FlowScheduledPathOffsetUnit (enumeration of type string)	Specify the time unit used to wait when the async action executes. Possible values are: • `Months` • `Days` • `Hours` • `Minutes`

Field Name	Field Type	Description
		This field is available in API version 61.0 and later.
`outputParameters`	FlowActionCallOutputParameter[]	An array of output parameters from the action to the flow.
`storeOutputAutomatically`	boolean	Indicates whether the action's output parameters are automatically available in the flow without creating any variables. When the value is `true`, you can reference an output parameter by specifying the API name of the Action element in the flow. The default value is `false`. When the value is `false`, create variables manually to store output values from the action. This field is available in API version 48.0 and later.
`timeoutConnector`	FlowConnector	Specifies which node to execute if an async action execution is timed out. This field is available in API version 62.0 and later.
`versionSegment`	int	Specifies the version of the versioned action. By default, the value is 1. Supported only when versionSegment is specified. This field is available in API version 58.0 to 61.0. This field is deprecated in API version 62.0 and later.
`versionString`	string	Reserved for future use.

FlowActionCallInputParameter

Defines an input parameter from the flow to the action. It extends FlowBaseElement and inherits all its fields. This metadata type is available in API version 31.0 and later.

Field Name	Field Type	Description
`name`	string	Required. Unique name for the input parameter.
`value`	FlowElementReferenceOrValue	Defines the value of the input parameter.

FlowActionCallOutputParameter

Defines an output parameter from the action to the flow. It extends FlowBaseElement and inherits all its fields. This metadata type is available in API version 31.0 and later.

Field Name	Field Type	Description
`assignToReference`	string	Required. Specifies the variable to which you want to assign the output parameter value.
`name`	string	Required. Unique name for the output parameter.

FlowApexPluginCall

Defines a call to an Apex plug-in from the flow. It extends FlowNode and inherits all its fields.

Field Name	Field Type	Description
`apexClass`	string	Required. The name of the Apex class.
`connector`	FlowConnector	Specifies which node to execute after this Apex plug-in call.
`faultConnector`	FlowConnector	Specifies which node to execute if the Apex plug-in call results in an error.
`inputParameters`	FlowApexPluginCallInputParameter[]	An array of input parameters from the flow to the Apex plug-in.
`outputParameters`	FlowApexPluginCallOutputParameter[]	An array of output parameters from the Apex plug-in to the flow.

FlowApexPluginCallInputParameter

Defines an input parameter from the flow to the Apex plug-in. It extends FlowBaseElement and inherits all its fields.

Field Name	Field Type	Description
`name`	string	Required. Unique name for the input parameter.
`value`	FlowElementReferenceOrValue	Defines the value of the input parameter.

FlowApexPluginCallOutputParameter

Defines an output parameter from the Apex plug-in to the flow. It extends FlowBaseElement and inherits all its fields.

Field Name	Field Type	Description
`assignToReference`	string	Required. Specifies the variable to which you want to assign the output parameter value.
`name`	string	Required. Unique name for the output parameter.

FlowAssignment

Defines an assignment node that can dynamically change the value of a variable in the flow. It extends FlowNode and inherits all of its fields.

Field Name	Field Type	Description
`assignmentItems`	FlowAssignmentItem[]	An array of assignment operations that's executed in the given order, starting from the index 0.
`connector`	FlowConnector	Specifies which node to execute after this assignment node.

FlowAssignmentItem

Defines an operation to apply to a variable. It extends FlowBaseElement and inherits all its fields.

Field Name	Field Type	Description
`assignToReference`	string	Reference to the variable to which you want to apply the specified operator.
`operator`	FlowAssignmentOperator (enumeration of type string)	Operation to apply to the variable reference in the `assignToReference` field. For valid values, see FlowAssignmentOperator.
`value`	FlowElementReferenceOrValue	Defines the value that you want the operator to apply to the variable reference in the `assignToReference` field.

FlowAssignmentOperator

An enumeration of type string that specifies the operation to apply to the variable in the `assignToReference` field. See "Flow Operators in Assignment Elements" in Salesforce Help.

These values are valid.

Enumeration Value	Description
`Add`	When the `assignToReference` field is a variable of type number or currency, this operator adds the `value` to the variable. When the `assignToReference` field is a variable of type date, this operator adds the `value` in days to the variable. When the `assignToReference` field is a variable of type string, this operator appends the `value` to the end of the string. When the `assignToReference` field is a variable of type picklist, this operator appends the `value` to the end of the last item in the picklist. When the `assignToReference` field is a variable of type multipicklist, this operator appends the `value` to the end of the last item in the multi-select picklist. To instead add an item to the end of the multi-select picklist, use the `AddItem` operator. When the `assignToReference` field is the `$Flow.ActiveStages` global variable, this operator appends the `value` as a new item at the end of `$Flow.ActiveStages`. When the `assignToReference` field is a collection variable, this operator appends the `value` to the end of the collection. Support for a collection variable as the `value` is available in API version 43.0 and later, but only via Metadata API. From Flow Builder, you can't save an Assignment element that contains a collection variable in the Value column for the `Add` operator. The `Add` operator isn't supported when the `assignToReference` field is a variable of type boolean, dateTime, or sObject.

Enumeration Value	Description
`AddAtStart`	Supported only when the `assignToReference` field is a collection variable or the `$Flow.ActiveStages` global variable. Adds the `value` as a new item at the beginning of the collection. When the `value` is a collection variable, the operator adds all items at the beginning of the collection. This operator is available in API version 43.0 and later.
`AddItem`	Supported only when the `assignToReference` field is a variable of type multipicklist. Adds the `value` to the picklist, including the semicolon that's required to mark a `value` as a separate item. This operator is available in API version 34.0 and later.
`Assign`	Assigns the `value` to the variable in the `assignToReference` field.
`AssignCount`	Supported only when the `value` is a collection variable or the `$Flow.ActiveStages` global variable. Counts the number of stages or items in the collection, and assigns that number to the variable in the `assignToReference` field. Corresponds to `equals count` in the user interface. This operator is available in API version 43.0 and later.
`RemoveAfterFirst`	Supported only when the `assignToReference` field is a collection variable or the `$Flow.ActiveStages` global variable. Finds the first instance of the `value` within the variable in the `assignToReference` field. Removes everything after that first instance from the variable. This operator is available in API version 43.0 and later.
`RemoveAll`	Supported only when the `assignToReference` field is a collection variable or the `$Flow.ActiveStages` global variable. Removes all instances of the `value` from the variable in the `assignToReference` field. When the `value` is a collection variable, the operator removes all instances of each item from the variable in the `assignToReference` field. This operator is available in API version 43.0 and later.
`RemoveBeforeFirst`	Supported only when the `assignToReference` field is a collection variable or the `$Flow.ActiveStages` global variable. Finds the first instance of the `value` within the variable in the `assignToReference` field. Removes everything before that first instance from the variable. This operator is available in API version 43.0 and later.
`RemoveFirst`	Supported only when the `assignToReference` field is a collection variable or the `$Flow.ActiveStages` global variable. Removes the first instance of the `value` from the variable in the `assignToReference` field. This operator is available in API version 43.0 and later.
`RemovePosition`	Supported only when the `assignToReference` field is a collection variable or the `$Flow.ActiveStages` global variable. Removes the item at the specified position. For example, if the collection contains three items, such as Red, Green, and Blue, and the `value` is 2, the second item, Green, is removed from the collection variable. This operator is available in API version 43.0 and later. Make sure that the `value` at run time is a positive integer within the range of the number of items in the collection variable.
`RemoveUncommon`	Supported only when `assignToReference` and `value` are both collection variables. Keeps items that are in both collections and removes the rest from the collection variable in the `assignToReference` field. This operator is available in API version 43.0 and later.

Enumeration Value	Description
`Subtract`	Supported only when the `assignToReference` field is a variable of type currency, date, or number. When the `assignToReference` field is a variable of type number or currency, this operator subtracts the `value` from the variable. When the `assignToReference` field is a variable of type date, this operator subtracts the `value` in days from the variable.

FlowBaseElement

Base class for all flow elements that require contextual information in metadata values. This class is an abstract class. FlowBaseElement is available in API version 32.0 and later.

Field Name	Field Type	Description
`processMetadataValues`	FlowMetadataValue[]	Contextual information for the element.

FlowChoice

A choice resource is a standalone choice option that you can reference or reuse throughout the flow. It extends FlowElement and inherits all of its fields.

Field Name	Field Type	Description
`choiceIcon`	FlowIcon	The icon to display for the choice in the screen. This field is available in API version 64.0 and later.
`choiceText`	string	Required. Choice label to display in the screen.
`dataType`	FlowDataType (enumeration of type string)	Required. Valid types are: • `Currency` • `Date` • `Number` • `String` • `Boolean`
`userInput`	FlowChoiceUserInput	Enables the choice to allow user input when the choice is selected. Not supported for choices in multi-select fields.
`value`	FlowElementReferenceOrValue	Actual value that's used during flow execution, for example, in assignments, calls to Apex plug-ins, and record elements. If null, this choice always has the value of null.

FlowChoiceUserInput

Allows the choice to include a user input field that appears when the user selects a choice. User input isn't supported for choices in multi-select fields. It extends FlowBaseElement and inherits all its fields.

Field Name	Field Type	Description
`isRequired`	boolean	Indicates whether users are required to enter something into the field when they select the choice.
`promptText`	string	Text that's displayed to prompt the user for input at runtime. Supports merge fields.
`validationRule`	FlowInputValidationRule	A rule used at runtime to validate the user input.

FlowCollectionProcessor

Defines a node that processes the contents of a collection, depending on the `collectionProcessorType`. FlowCollectionProcessor is available in API version 50.0 and later. FlowCollectionProcessor extends FlowNode and inherits all its fields.

Field Name	Field Type	Description
`assignNextValueToReference`	string	The name of the variable that's assigned to the next value of the collection.
`collectionProcessorType`	FlowCollectionProcessorType	The type of the collection processor. Valid values are: • `SortCollectionProcessor`—This value is available in API version 50.0 and later. • `RecommendationMapCollectionProcessor`—This value is available in API version 53.0 and later. • `FilterCollectionProcessor`— This value is available in API version 53.0 and later.
`collectionReference`	string	The collection being sorted, filtered, or assigned to recommendations.
`conditionLogic`	string	Defines how the filtering conditions are evaluated. Valid values are: • `And` • `Or` • `Custom logic, such as (1 AND (2 OR 3))` • `Formula`
`conditions`	FlowCondition[]	An array of conditions for the input collection.
`connector`	FlowConnector	Specifies which node to execute after processing the collection.

Field Name	Field Type	Description
`formula`	string	The formula expression that filters the input collection. If the formula evaluates to `true`, the record is added to the output collection.
`limit`	int	The maximum number of records to include in the generated collection. There's no default value. All items of the collection are kept if it's greater than the size of the collection. If sortField and sortOrder are also specified, the records are sorted before the limit takes effect. This field is available in API version 51.0 and later. This field is nillable in API version 51.0 and later.
`mapItems`	FlowCollectionMapItem[]	The rules to map each field of the collection variable.
`outputSObjectType`	string	The sObject type of the output collection.
`sortOptions`	FlowCollectionSortOption[]	An array of options to sort the items in the collection. This field is available in API version 51.0 and later.

FlowCollectionSortOption

Sets the sorting field, sort order, and placement of empty or null values in the sorted collection. This metadata type is available in API version 51.0 and later.

Field Name	Field Type	Description
`doesPutEmptyStringAndNullFirst`	boolean	Place empty or null values first in the sorted list by setting this value to `true`. The default value is `false`.
`sortField`	string	Determines the sorting of records that meet the filter criteria. Required for record collections and collections of Apex-defined variables. If the collection is a primitive data type, such as a list of string or integer values, `sortField` isn't supported.
`sortOrder`	SortOrder (enumeration of type string)	The order that the collection is sorted in. Valid values are: • `Asc`—Ascending • `Desc`—Descending

FlowCustomError

Defines a custom error element to roll back a change that triggered a flow and inform the user exactly what caused the error. It extends FlowNode and inherits all its fields.

Field Name	Field Type	Description
`description`	string	Describes the error message.
`connector`	FlowConnector	Required. Which node to execute after completing the current node.
`customErrorMessages`	FlowCustomErrorMessage[]	An array of custom error messages.

FlowCustomErrorMessage

Defines a custom error message for a custom error element. It extends FlowBaseElement and inherits all its fields.

Field Name	Field Type	Description
`errorMessage`	string	Required. Specifies the custom error message.
`fieldSelection`	string	References the erroneous field that's associated with the custom error message.
`isFieldError`	boolean	Required. When this field is set to `true`, indicates that the custom error message displays inline on a field. When it is set to `false`, it displays in a window on a record page. The default value is `false`.

FlowCondition

Defines a condition for a rule. It extends FlowBaseElement and inherits all its fields.

Field Name	Field Type	Description
`aggregationOperator`	string	Operation to apply to the variable reference in the `assignToReference` field. The valid value is: • `Count`
`conditionLogic`	string	Specifies logic for the conditions. Value can be: • `and`—Evaluates to `true` only if all its conditions evaluate to true • `or`—Evaluates to `true` if any of its conditions evaluate to true
`conditionType`	FlowWaitConditionType (enumeration of type string)	The type of condition that a requirement in an automation is used for. Valid values are: • Container • EntryCondition • ExitCondition

Field Name	Field Type	Description
`conditions`	FlowCondition[]	An array of conditions that must be `true` for the flow to execute the rule.
`leftValueReference`	string	Required. Unique name of the element that serves as the left side of the condition expression.
`operator`	FlowComparisonOperator (enumeration of type string)	Required. Comparison operators in conditions for flow elements and resources. Valid values are: • `Contains` • `EndsWith` • `EqualTo` • `GreaterThan` • `GreaterThanOrEqualTo` • `HasError`—This value is available in API version 64.0 and later. • `In`— This value is available in API version 56.0 and later. • `IsBlank`—A text value with zero characters or with only whitespace. Use to determine whether a text field or variable is blank. For other data type values, use to determine whether a field or variable is null. This value is available in API version 61.0 and later. • `IsChanged`– This value is available in API version 52.0 and later. • `IsEmpty`—An empty collection. This value is available in API version 61.0 and later. • `IsNull`—A value that is either not set or references no value. Use to determine whether a field or variable value is set to no value. • `LessThan` • `LessThanOrEqualTo` • `None`— Save a flow with an incomplete condition, so you can finish building the flow later. This value is available in API version 58.0 and later. • `NotEqualTo` • `NotIn`— This value is available in API version 56.0 and later. • `StartsWith` • `WasSelected`— Requires a choice on the left side. • `WasSet`— This value is available in API version 30.0 and later. • `WasVisited`— Requires a node on the left side. See Flow Operators.

Field Name	Field Type	Description
`rightValue`	FlowElementReferenceOrValue	Unique name of an element or the actual value, such as text or a number, for the right side of the condition expression.

FlowCustomProperty

Defines the name and value of a custom property in a flow. This metadata type is available in API version 63.0 and later.

Field Name	Field Type	Description
`name`	string	Required. Unique name for the custom property associated with the flow.
`value`	FlowElementReferenceOrValue	Defines the value of the custom property associated with the flow. When the FlowCustomProperty's name is set to `ScreenProgressIndicator`, valid values are: • `"Location":"Top","Type":"Simple"` • `"Location":"Footer","Type":"Simple"` • `"Location":"Top","Type":"Path"`

FlowActionCallPath

A path determines which node of the flow is executed after the Einstein Decision element. A path defines and links to the subsequent node. It extends FlowBaseElement and inherits all its fields.

Field Name	Field Type	Description
`connector`	FlowConnector	Required. Which node to execute after completing the current node.
`pathName`	string	Required. Unique name for the path.

FlowConnector

Connectors determine the order in which the nodes of the flow are executed. A connector defines and links to the subsequent node. It extends FlowBaseElement and inherits all its fields.

Field Name	Field Type	Description
`isGoTo`	boolean	Make the connector a Go To Connector by setting this value to `true`. The default value is `false`. This value is available in API version 53.0 and later. See Flow Connectors.
`targetReference`	string	Required. Which node to execute after completing the current node.

FlowCollectionMapItem

Defines the rule to assign a value to the field reference. This metadata type is available in API version 51.0 and later.

Field Name	Field Type	Description
`assignToFieldReference`	string	Required. Specifies the reference to the field to which the specified operator is applied.
`operator`	FlowAssignmentOperator (enumeration of type string)	Required. Applies to the variable reference in the `assignToFieldReference` field.
`value`	FlowElementReferenceOrValue	Required. Defines the value that the operator applies to the variable reference in the `assignToFieldReference` field.

FlowDataTypeMapping

This data type mapping defines the specific sObject data type for input and out values that have the generic sObject data type. It extends FlowBaseElement and inherits all its fields. This metadata type is available in API version 48.0 and later.

Field Name	Field Type	Description
`apexClass`	string	The name of the Apex class. This field is available in API version 61.0 and later.
`typeName`	string	Required. API name of the input or output variable. The `T__` prefix is required for input variables. The `U__` prefix is required for output variables. For example, `T__inputCollection` represents the API name of the input variable `inputCollection`.
`typeValue`	string	API name of the specific sObject data type that this value maps to. For example, `Account`.

FlowConstant

A constant resource defines a fixed value that can be used throughout your flow. It extends FlowElement and inherits all its fields.

Field Name	Field Type	Description
`dataType`	FlowDataType (enumeration of type string)	Required. Valid types are: • `Currency` • `Date` • `Number` • `String`

Field Name	Field Type	Description
		• `Boolean`
`value`	FlowElementReferenceOrValue	Default value of the constant. This field can't have merge fields, nor can it reference another resource besides `$GlobalConstant.EmptyString`.

FlowDecision

A node that evaluates a set of rules and routes the flow execution based on the first rule that evaluates to true. It extends FlowNode and inherits all its fields.

Field Name	Field Type	Description
`defaultConnector`	FlowConnector	Specifies which node to execute if none of the rules evaluate to true.
`defaultConnectorLabel`	string	Label for the default connector.
`rules`	FlowRule[]	An array of rules for the decision. The rules are evaluated in the order that they're listed, and the connector of the first true rule is used. If no rules are true, then the default connector is used. In Flow Builder, rules are referred to as decision outcomes.

FlowDynamicChoiceSet

Retrieves data or metadata from an object and dynamically generates a set of choices at run time. It extends FlowElement and inherits all its fields. Depending on the fields that are set, this element represents a record choice or a picklist choice.

- A *record choice* dynamically generates choices based on records that meet specified filter criteria. If a dynamic choice doesn't have the `picklistField` and `picklistObject` parameters set, it's a record choice and it can't have a data type of `Picklist` or `Multipicklist`.
- A *picklist choice* dynamically generates choices based on the available values for a picklist or multi-select picklist field. If a dynamic choice has the `picklistField` and `picklistObject` parameters set, it's a picklist choice and it must have a data type of `Picklist` or `Multipicklist`.

Field Name	Field Type	Description
`collectionReference`	string	The collection that's used to generate choices. This field is available in API version 54.0 and later.
`dataType`	FlowDataType (enumeration of type string)	Required. Valid types are: • `Boolean` • `Currency` • `Date` • `Multipicklist`—Picklist choices only • `Number`

Field Name	Field Type	Description
		• `Picklist`—Picklist choices only • `Record` • `String` `Picklist` and `Multipicklist` are available in API version 35.0 and later. `Record` is available in API version 54.0 and later.
`displayField`	string	Required for record choices. Specifies the object field. The values of the object field are displayed to the user as choice labels for selecting a record. For example, for an account, if you want the dynamically generated choices to be displayed as the account names from the records that are retrieved from the database, specify `Name` in `displayField`. Not supported for picklist choices. Picklist choices always display the labels for the retrieved picklist values.
`filters`	FlowRecordFilter[]	An array of filters to apply to the records retrieved from the database. For example, filter accounts to include only the accounts that were created in the past three months. Not supported for picklist choices.
`limit`	int	Maximum number of choices to include in the generated set of choices. Maximum and default: 200. If `sortField` and `sortOrder` are also specified, the records are sorted before the `limit` takes effect. This field is available in API version 25.0 and later. This field is nillable in API version 45.0 and later.
`object`	string	Required for record choices. The object whose fields you want to retrieve from the database and use to generate the set of choices. For example, use "Account" to dynamically generate choices from the information in account records in the database. Not supported for picklist choices.
`outputAssignments`	FlowOutputFieldAssignment[]	An array that assigns fields from the user-selected record to variables that can be used elsewhere in the flow. For example, when the user selects an account name from the dynamically generated list of choice options, outputAssignments can assign the ID and AnnualRevenue from the user-selected account to variables that you specify. Not supported for picklist choices.

Field Name	Field Type	Description
`picklistField`	string	Required for picklist choices. The field whose available values you want to retrieve from the database and use to generate the picklist choice. For example, use "Industry" to dynamically generate one choice for each available value on the Industry picklist field. Not supported for record choices. This field is available in API version 35.0 and later.
`picklistObject`	string	Required for picklist choices. The object whose field metadata you want to retrieve from the database and use to generate the picklist choice. For example, use "Account" to dynamically generate choices from a picklist field on the Account object. Not supported for record choices. This field is available in API version 35.0 and later.
`sortField`	string	Field that's used for sorting records that meet the filter criteria. If this field isn't specified, the returned records aren't sorted. You can only sort records by fields that have the `Sort` API field property, as specified in SOAP API. Not supported for picklist choices. This field is available in API version 25.0 and later.
`sortOrder`	SortOrder (enumeration of type string)	Order in which to sort the records. If this field isn't specified, then the results aren't sorted. Valid values are: • `Asc`—Ascending • `Desc`—Descending Not supported for picklist choices. This field is available in API version 25.0 and later.
`valueField`	string	Stored value for the choice, which can differ from what is displayed to the user as the choice options (`displayField`). For example, the `displayField` could be the account "Name" while the valueField is the account "Id." Not supported for picklist choices. Picklist choices always store the API value for the retrieved picklist values.

FlowElement

Base class for all flow elements. This class is an abstract class. It extends FlowBaseElement and inherits all its fields.

Field Name	Field Type	Description
`description`	string	Description of the flow element.
`name`	string	Unique name of the flow element.

FlowElementReferenceOrValue

Defines a reference to an existing element or a particular value that you specify. Make sure that you specify only *one* of the fields.

Field Name	Field Type	Description
`apexValue`	string	Use this field to specify a JSON response value of an Apex-defined record. Use this field only for `FlowScreenFieldInputParameter` and `FlowActionCallInputParameter`. If you want to specify a different data type or element reference, don't use this field.
`booleanValue`	boolean	Use this field to specify a boolean value. If you want to specify a different data type or element reference, don't use this field.
`complexValue`	string	When `complexValueType` is specified, use this field to specify flow resources and fields in the data structure. Use these fields to describe the data structure: • `fieldReference`—The list of field API names. • `objectType`—The object type, sObject or Apex. • `type`—The API name of the sObject or Apex class. • `elementReference`—The API name of the flow resource that contains the list of fields specified in `fieldReference`. This field is available in API version 63.0 and later.
`complexValueType`	FlowComplexValueType (enumeration of type string)	Use this field to specify the type of data structure to reference. Valid values are: • `JoinDefinition`—When `InnerJoin` is specified in `transformType`, indicates flow resources for source and target collections, join keys, selected fields to join. and field mappings in a join transformation. • `FieldReference`—Use this field to define the flow resource and its fields referenced in the flow. This field is available in API version 63.0 and later. Use `complexValue` to specify the data structure.
`dateTimeValue`	dateTime	Use this field to specify a dateTime value. If you want to specify a different data type or element reference, don't use this field. This field is available in API version 30.0 and later.

Field Name	Field Type	Description
`dateValue`	date	Use this field to specify a date value. If you want to specify a different data type or element reference, don't use this field.
`elementReference`	string	Use this field to specify the name of an existing flow resource. If you want to specify a value instead of an element reference, don't use this field.
`formulaDataType`	FlowDataType (enumeration of type string)	Use this field to specify the formula result's data type of the transformed data. Corresponds to the target data field in Flow Builder. This field requires the `formulaExpression` field. This field is available in API version 59.0 and later. See FlowTransform Valid values are: • `Apex` • `Boolean` • `Currency` • `Date` • `DateTime` • `Number` • `String` • `sObject`—This value corresponds to a record variable.
`formulaExpression`	string	Use this field to specify the formula expression that transforms the data in the flow. In Flow Builder, it corresponds to the target data field in the Transform element. This field requires the `formulaDataType` field. This field is available in API version 59.0 and later. See FlowTransform.
`numberValue`	double	Use this field to specify a double value. If you want to specify a different data type or element reference, don't use this field.
`setupReference`	string	Use this field to specify the name of an existing setup reference. Required for Omni-Channel elements. If you want to specify a value instead of a setup reference, don't use this field. Required when `setupReferenceType` is specified.
`setupReferenceType`	string	Use this field to specify the type of setup reference. Required when setupReference is specified.
`sobjectValue`	string	Use this field to specify a JSON response value of an sObject record. Use this field only for `FlowScreenFieldInputParameter` and `FlowActionCallInputParameter`. If you want to specify a different data type or element reference, don't use this field.
`stringValue`	string	Use this field to specify a string value. If you want to specify a different data type or element reference, don't use this field. When the FlowMetadataValue's `name` field is set to `SendNoApproverEmails`, valid values are `true` or `false` and are case-insensitive.

Field Name	Field Type	Description
		When the FlowMetadataValue's `name` field is set to `BuilderType` or `OriginalBuilderType`, the valid value is `LightningFlowBuilder`. The value is reserved for internal use.
`transform`	FlowInlineTransform	Use this field to specify a value for an inline data transformation. This field is available in API version 62.0 and later.
`transformValueReference`	string	Reserved for future use.

FlowExitRule

Defines the conditions and logic that enables an exit rule to evaluate to true. It extends FlowElement and inherits all of its fields.

Field Name	Field Type	Description
`conditions`	FlowCondition[]	An array of conditions for the exit rule.
`label`	string	Required. Label for the exit rule.
`logicalOperator`	string	Required. Logical operator in conditions for the exit rule. Valid values are: • `and`—Evaluates to `true` only if all its conditions evaluate to true • `or`—Evaluates to `true` if any of its conditions evaluate to true
`ruleOrder`	int	Indicates how the exit rule is ordered against other exit rules. The `ruleOrder` value must be unique within the flow.

FlowExperiment

A node that routes the flow execution based on a specified experiment distribution percentage. It extends FlowNode and inherits all its fields. This metadata type is available in API version 61.0 and later.

Field Name	Field Type	Description
`duration`	int	The amount of time that the experiment runs. This field is available in API version 64.0 and later.
`durationUnit`	string	The unit of measurement for experiment duration. Valid values are: • Minutes • Hours • Days • Weeks • Months

Field Name	Field Type	Description
		This field is available in API version 64.0 and later.
`paths`	FlowExperimentPath[]	An array of flow experiment paths.
`testGroupPercentage`	int	Specifies the distribution percentage of the test group. A valid number in the range 0-99. This field is available in API version 64.0 and later.
`type`	FlowExperimentType	Required. The type of experiment. Valid value is: • Random

FlowExperimentPath

Defines an experiment path. It extends FlowElement and inherits all its fields. This metadata type is available in API version 61.0 and later.

Field Name	Field Type	Description
`connector`	FlowConnector	Specifies which node to execute after this experiment path.
`label`	string	Required. Label for the path.
`percentage`	int	Required. The distribution percentage for this path.

FlowFormula

Calculates a value using functions and elements in the flow. It extends FlowElement and inherits all its fields.

Field Name	Field Type	Description
`dataType`	FlowDataType (enumeration of type string)	The data type for the formula. Valid values are: • `Boolean` • `Currency` • `Date` • `DateTime` • `Number` • `String` `dataType` defaults to `Number` if it isn't defined in a formula. This field is available in API version 31.0 and later.
`expression`	string	Required. Salesforce formula expression. The return value must match the data type. For API version 30.0 and earlier, the return value must be numeric.
`scale`	int	Scale of the return value, specifically, the number of digits to the right of the decimal point. Available only when the data type is

Field Name	Field Type	Description
		Number or Currency. Corresponds to the Decimal Places field in Flow Builder.

FlowIcon

Allows a resource to include an icon. This metadata type is available in API version 64.0 and later.

Field Name	Field Type	Description
`iconName`	String	The name of the selected Salesforce Lightning Design System icon. This field is available in API version 64.0 and later.

FlowInputFieldAssignment

Assigns the value for a record field based on a resource or static value. It extends FlowBaseElement and inherits all its fields.

Field Name	Field Type	Description
`field`	string	Required. The name of the field to assign a value to when a record is created or updated.
`value`	FlowElementReferenceOrValue	The value to assign to the field.

FlowInputValidationRule

Validation rules verify that the data entered by the user meets the specified requirements. If the validation rule evaluates to false, then the specified error message is displayed.

Field Name	Field Type	Description
`errorMessage`	string	Required. The error message to display when `formulaExpression` is `false`.
`formulaExpression`	string	Required. A formula that's used to validate the user input.

FlowLoop

A construct for iterating through a collection. It extends FlowNode and inherits all its fields. FlowLoop is available in API version 30.0 and later.

Field Name	Field Type	Description
`assignNextValueToReference`	string	The variable that's assigned to the current value in the collection before navigating to the target of `nextValueConnector`.
`collectionReference`	string	The collection being looped through.

Field Name	Field Type	Description
`iterationOrder`	iterationOrder (enumeration of type string)	Valid values are: • `Asc`—Iterate through the collection in the order the values are listed (first to last). • `Desc`—Iterate through the collection in the reverse order the values are listed (last to first).
`nextValueConnector`	FlowConnector	A reference to the next element in the collection.
`noMoreValuesConnector`	FlowConnector	The element to navigate to when all entries in the collection have been iterated through.

FlowMetadataValue

Defines contextual information that can be passed between elements in a flow. Flow metadata values can be used in an application that produces or consumes flows. FlowMetadataValue is available in API version 31.0 and later.

Field Name	Field Type	Description
`name`	string	Required. Name for the metadata value. This name doesn't need to be unique across all elements. To specify that an approval orchestration send no email notifications to approvers, use SendNoApproverEmails .
`value`	FlowElementReferenceOrValue	Reference or value for the metadata value.

FlowNode

A node is a type of element that's visible in the flow diagram. It extends FlowElement and inherits all its fields.

Field Name	Field Type	Description
`elementSubtype`	FlowElementSubtype (enumeration of type string)	Reserved for internal use.
`label`	string	Name of the node. This non-unique label is different from the unique name of the node, which is inherited from FlowElement.
`locationX`	int	Required. Horizontal location of the node, in pixels from the left. In API version 64.0 and later, if a flow is saved in auto-layout, this field is set to 0.
`locationY`	int	Required. Vertical location of the node, in pixels from the top. In API version 64.0 and later, if a flow is saved in auto-layout, this field is set to 0.

FlowOrchestratedStage

A stage node that contains steps in an orchestration. It extends FlowNode and inherits all its fields. This metadata type is available in API version 53.0 and later.

Field Name	Field Type	Description
`connector`	FlowConnector	Specifies which node to execute after this stage.
`exitActionInputParameters`	FlowStageStepExitActionInputParameter[]	An array of input parameters from the stage to the evaluation flow. These parameters specify an exit condition for the stage.
`exitActionName`	string	The name of the evaluation flow used as an exit condition for the stage.
`exitActionOutputParameters`	FlowStageStepExitActionOutputParameter[]	An array of output parameters from the evaluation flow to the stage. These parameters specify an exit condition for the stage.
`exitActionType`	InvocableActionType (enumeration of type string)	The type of the evaluation flow for the custom exit condition. Valid values are: • `createFieldGnrnPromptTmplResp` • `EvaluationFlow` • `exploreConversation` This value is available in API version 61.0 and later.
`exitConditionLogic`	string	Defines how the stage exit conditions are evaluated. Valid values are: • `And` • `Or` • `Custom logic, such as (1 AND (2 OR 3))` • `Formula`
`exitConditions`	FlowCondition[]	An array of requirements that must be met to exit the stage.
`faultConnector`	FlowConnector	Not used.
`runAsUser`	boolean	Indicates whether an asynchronous background step is run in the context of the user who completed the most recently completed interactive step.
`stageSteps`	FlowStageStep[]	An array of stage step resources.

FlowOutputFieldAssignment

Assigns a record field's value to a variable that can be used elsewhere in the flow. It extends FlowBaseElement and inherits all its fields.

Field Name	Field Type	Description
`assignToReference`	string	Required. Reference to the variable where you want to store the value of the record field.
`field`	string	Required. Name of the field whose value is to be assigned after a record lookup.

FlowRelatedRecordLookup (Beta)

Note: This feature is a pilot or beta service that is subject to the Beta Services Terms at Agreements - Salesforce.com or a written Unified Pilot Agreement if executed by Customer, and applicable terms in the Product Terms Directory. Use of this pilot or beta service is at the Customer's sole discretion.

Finds records in the database that are related to the records specified in FlowRecordLookup and stores their field values in the flow. Corresponds to a Get Records element in Flow Builder. It extends FlowBaseElement and inherits all its fields.

Field Name	Field Type	Description
`filterLogic`	string	The filter logic that's applied to the filter condition requirements. To require all conditions, use `AND`. To require any conditions, use `OR`. For custom condition logic, enter the entire logic string. For example, `1 AND 2 OR (3 AND 4)`.
`filters`	FlowRecordFilter[]	An array that specifies the criteria used to select the record from the database. If the filters return more than one record, they're sorted according to the specified `sortField` and `sortOrder`. If `outputReference` specifies a non-collection record variable or if `getFirstRecordOnly` is `true`, only the first record in the sorted list is selected. If `sortField` or `sortOrder` isn't specified, records aren't returned in any particular order. If `outputReference` specifies a non-collection record variable or if `getFirstRecordOnly` is `true`, only the first record in the unsorted list is selected.
`getFirstRecordOnly`	boolean	Indicates whether to store field values for only one record, even when

Field Name	Field Type	Description
		multiple records meet the filter criteria. Supported only when `storeOutputAutomatically` is `true`. When `storeOutputAutomatically` is `false`, what determines whether one or multiple records are stored is whether `outputReference` specifies a record variable or a record collection variable.
`limit`	FlowElementReferenceOrValue	Specifies the maximum number of records to store. Valid values are between 2 and 20,000. Supported only when `getFirstRecordOnly` is false.
`queriedFields`	strings[]	An array that specifies which fields from the selected record are saved to the specified record variable.
`relatedObject`	string	Name of the related object from which to select related records.
`relatedRecords`	FlowRelatedRecordLookup[]	An array that specifies the related records to look up in the database.
`relationshipField`	string	Specifies the API name of the relationship field used to link the object to its related object. This field is required for retrieving related records.
`sortField`	string	The field that's used for sorting the records that meet the filter criteria. If this field isn't specified, the returned records aren't sorted. You can only sort records by fields that have the `Sort` API field property, as specified in SOAP API.
`sortOrder`	SortOrder (enumeration of type string)	Order in which to sort the records. If this field isn't specified, then the results aren't sorted. Valid values are: • `Asc`—Ascending • `Desc`—Descending

FlowRecordCreate

Create a record in the database using values from the flow. It extends FlowNode and inherits all its properties.

Note: The flow record `create`, `lookup`, `update`, and `delete` operations are different from the CRUD-based metadata calls `create()`, `retrieve()`, `update()`, and `delete()`. The flow record methods apply to record operations from within a flow, which aren't the same as doing any metadata calls to CRUD setup entities.

Field Name	Field Type	Description
`assignRecordIdToReference`	string	Reference to the variable where you want to store the ID after the record is created.
`connector`	FlowConnector	Specifies which node to execute after creating the record.
`doesUpsert`	boolean	Indicates whether the element creates or updates records. The default value is `false`, indicating that the element only creates records. This field is available in API version 62.0 and later.
`doesUpsertAllOrNone`	boolean	Indicates whether the element creates or updates records only if all records are created or updated successfully. If set to `true` and a record fails, then the transaction rolls back and no records are created or updated. If set to `false`, the transaction creates or updates only the records that are successful. The default value is `true`. This field is available in API version 62.0 and later.
`faultConnector`	FlowConnector	Specifies which node to execute if the attempt to create a record results in an error.
`filterLogic`	string	The filter logic applied to the filter condition requirements. To require all conditions, use `AND`. To require any conditions, use `OR`. For custom condition logic, enter the entire logic string. For example, `1 AND 2 OR (3 AND 4)`. This field is available in API version 61.0 and later.
`filters`	FlowRecordFilter[]	An array that specifies the criteria to select which records to create or update in the database. This field is available in API version 61.0 and later.
`inputAssignments`	FlowInputFieldAssignment[]	An array that assigns values to the specified fields of the record being created.
`inputReference`	string	Specifies the record variable whose field values are used to populate the new record's fields.
`object`	string	Required. The object type that the element creates.

Field Name	Field Type	Description
operationMultMatchingRecords	string	The operation to perform if multiple matching records are found. Valid values are: • None • UpdateAllRecords • UpdateLatestRecord This field is available in API version 61.0 and later.
operationOneMatchingRecord	string	The operation to perform if one matching record is found. Valid values are: • None • UpdateAllRecords This field is available in API version 61.0 and later.
operationZeroMatchingRecords	string	The operation to perform if no matching records are found. Valid values are: • None This field is available in API version 61.0 and later.
storeOutputAutomatically	boolean	Indicates whether the record ID is automatically available in the flow without creating any variables. When the value is true, you can reference the record ID by specifying the API name of the Create Records element in the flow. The default value is false. When the value is false, create a variable to store the record ID. This field is available in API version 48.0 and later.
upsertExternalIdField	string	If doesUpsert is true, specifies the external ID field on the record. You can provide a value for this property or for the Upsert Standard ID Field property, but not both. This field is available in API version 62.0 and later.
upsertStandardIdField	string	If doesUpsert is true, specifies the standard ID field like Account ID on the object. You can provide a value for this property or for the Upsert External ID Field property, but not both. This field is available in API version 62.0 and later.

FlowRecordDelete

Deletes one or more records in the database. It extends FlowNode and inherits all its fields.

Note: The flow record create, lookup, update, and delete operations are different from the CRUD-based metadata calls create(), retrieve(), update(), and delete(). The flow record methods apply to record operations from within a flow, which aren't the same as doing any metadata calls to CRUD setup entities.

Field Name	Field Type	Description
`connector`	FlowConnector	Specifies which node to execute after deleting the record.
`faultConnector`	FlowConnector	Specifies which node to execute if the attempt to delete a record results in an error.
`filters`	FlowRecordFilter[]	An array that specifies the criteria used to select which records to delete from the database. For example, delete accounts whose last activity was older than a specified date.
`inputReference`	string	Specifies the record variable whose record ID is used to identify which record to delete in the database.
`object`	string	Required. The name of the object whose records are deleted.

FlowRecordFilter

Sets the criteria for searching records in the database. It extends FlowBaseElement and inherits all its fields.

Field Name	Field Type	Description
`field`	string	Required. The field to be used for filtering records.
`operator`	FlowRecordFilterOperator (enumeration of type string)	Required. Valid values are: • `EqualTo` • `NotEqualTo` • `GreaterThan` • `LessThan` • `GreaterThanOrEqualTo` • `LessThanOrEqualTo` • `StartsWith` • `EndsWith` • `Contains` • `IsNull`
`value`	FlowElementReferenceOrValue	Reference or value used with the field and operator to filter records.

FlowRecordLookup

Finds records in the database and stores their field values in the flow. Corresponds to a Get Records element in Flow Builder. It extends FlowNode and inherits all its fields.

Note: The flow record `create`, `lookup`, `update`, and `delete` operations are different from the CRUD-based metadata calls `create()`, `retrieve()`, `update()`, and `delete()`. The flow record methods apply to record operations from within a flow, which aren't the same as doing any metadata calls to CRUD setup entities.

Field Name	Field Type	Description
`assignNullValuesIfNoRecordsFound`	boolean	Specifies that all values are set to `null` when no record is found. Supported only when `storeOutputAutomatically` is `false`. This field is available in API version 30.0 and later.
`connector`	FlowConnector	Specifies which node to execute after getting records from the database.
`faultConnector`	FlowConnector	Specifies which node to execute if the attempt to get records results in an error.
`filterLogic`	string	The filter logic that's applied to the filter condition requirements. To require all conditions, use `AND`. To require any conditions, use `OR`. For custom condition logic, enter the entire logic string. For example, `1 AND 2 OR (3 AND 4)`. This field is available in API version 50.0 and later.
`filters`	FlowRecordFilter[]	An array that specifies the criteria used to select the record from the database. If the filters return more than one record, they're sorted according to the specified `sortField` and `sortOrder`. If `outputReference` specifies a non-collection record variable or if `getFirstRecordOnly` is `true`, only the first record in the sorted list is selected. If `sortField` or `sortOrder` isn't specified, records aren't returned in any particular order. If `outputReference` specifies a non-collection record variable or if `getFirstRecordOnly` is `true`, only the first record in the unsorted list is selected.
`getFirstRecordOnly`	boolean	Indicates whether to store field values for only one record, even when multiple records meet the filter criteria. Supported only when

Field Name	Field Type	Description
		`storeOutputAutomatically` is `true`. When `storeOutputAutomatically` is `false`, what determines whether one or multiple records are stored is whether `outputReference` specifies a record variable or a record collection variable. This field is available in API version 47.0 and later.
`limit`	FlowElementReferenceOrValue	Specifies the maximum number of records to store. Valid values are between 2 and 20,000. Supported only when `getFirstRecordOnly` is false. This field is available in API version 63.0 and later.
`object`	string	Name of the object from which to select the record.
`outputAssignments`	FlowOutputFieldAssignment[]	An array that assigns fields from the selected record to variables that can be used elsewhere in the flow. Supported only when `storeOutputAutomatically` is `false`.
`outputReference`	string	Specifies the record variable or record collection variable that stores the queried fields' values. Supported only when `storeOutputAutomatically` is `false`.
`queriedFields`	string[]	An array that specifies which fields from the selected record are saved to the specified record variable.
`relatedRecords` (beta)	FlowRelatedRecordLookup[]	An array that specifies the related records to look up in the database.
`sortField`	string	The field that's used for sorting the records that meet the filter criteria. If this field isn't specified, the returned records aren't sorted.

Field Name	Field Type	Description
		You can only sort records by fields that have the `Sort` API field property, as specified in SOAP API. This field is available in API version 25.0 and later.
`sortOrder`	SortOrder (enumeration of type string)	Order in which to sort the records. If this field isn't specified, then the results aren't sorted. Valid values are: • `Asc`—Ascending • `Desc`—Descending This field is available in API version 25.0 and later.
`storeOutputAutomatically`	boolean	Indicates whether the returned records' field values are automatically available in the flow without creating any variables. When the value is `true`, the flow can reference a field by specifying the `name` of the Get Records element and the record field, such as `Get_Contacts.AccountId`. Supported only when `processType` is `Flow` or `AutoLaunchedFlow`. This field is available in API version 47.0 and later.

FlowRecordRollback

Rolls back the current transaction and cancels its pending record changes. Corresponds to the Roll Back Records element in Flow Builder. Available only in screen flows.

FlowRecordRollback extends FlowNode and inherits all its fields. This metadata type is available in API version 52.0 and later.

Field Name	Field Type	Description
`connector`	FlowConnector	Specifies which node to execute after rolling back the current transaction.

FlowRecordUpdate

Finds records in the database and updates them with values from the flow. It extends FlowNode and inherits all its fields.

Note: The flow record `create`, `lookup`, `update`, and `delete` operations are different from the CRUD-based metadata calls `create()`, `retrieve()`, `update()`, and `delete()`. The flow record methods apply to record operations from within a flow, which aren't the same as doing any metadata calls to CRUD setup entities.

Field Name	Field Type	Description
`connector`	FlowConnector	Specifies which node to execute after completing the record update.
`faultConnector`	FlowConnector	Specifies which node to execute if the attempt to update a record results in an error.
`filters`	FlowRecordFilter[]	An array that specifies the criteria used to select the records to update in the database.
`inputAssignments`	FlowInputFieldAssignment[]	An array that assigns values to the specified fields of the record being updated.
`inputReference`	string	Specifies the record variable whose field values are used to update the record's fields.
`object`	string	Required. Name of the object whose records are updated.

FlowRule

Defines the conditions and logic that enables a rule to evaluate to true. It extends FlowElement and inherits all of its fields.

Field Name	Field Type	Description
`conditionLogic`	string	Specifies logic for the conditions. Value can be: • `and`—Evaluates to `true` if all of its conditions are true. • `or`—Evaluates to `true` if any conditions are true. • Advanced logic like `1 AND (2 OR 3)`—Evaluates to `true` if the first condition is `true` and either the second or third condition is `true`. When you use advanced logic, the string can contain up to 1,000 characters.
`conditions`	FlowCondition[]	An array of conditions for the rule.
`connector`	FlowConnector	Specifies which node to execute if this rule evaluates to `true` in a decision first.
`doesRequireRecordChangedToMeetCriteria`	boolean	If set to `true`, conditions evaluate to `true` only if the record didn't meet the required conditions before the triggering update but now meets the conditions after the update. This field is available in API version 50.0 and later.
`label`	string	Required. Label for the connector.

FlowSchedule

Specifies when and how frequently to run the flow. This metadata type is available in API version 47.0 and later.

Field Name	Field Type	Description
`frequency`	FlowStartFrequency (enumeration of type string)	Specifies how frequently to run the flow. Valid values are: • `Once` • `Daily` • `Weekly` • `OnActivate`—This value is available in API version 49.0 and later.
`startDate`	date	The date when the flow runs, or when the flow's run schedule starts recurring.
`startTime`	time	The time of day when the flow runs, based on the org's default time zone.

FlowScheduledPath

Defines a scheduled path. It extends FlowElement and inherits all its fields. This metadata type is available in API version 51.0 and later.

Field Name	Field Type	Description
`connector`	FlowConnector	Specifies which node to execute after this scheduled path.
`label`	string	Label for the scheduled path.
`maxBatchSize`	int	The maximum number of scheduled path interviews to execute in a single batch, from `1` to `200`. Default is `200`.
`offsetNumber`	int	Number of months, days, hours, or minutes to offset the time that the scheduled path executes. Negative values offset the time to execute before the provided time. Positive values offset the time to execute after the provided time.
`offsetUnit`	FlowScheduledPathOffsetUnit (enumeration of type string)	Specify the time unit used to offset when the scheduled path executes. Possible values are: • `Months`—This value is available in API version 56.0 and later. • `Days` • `Hours` • `Minutes`
`pathType`	FlowScheduledPathType (enumeration of type string)	The type of scheduled path. `null` is used for time-triggered and record-triggered paths. The default value is `null`. • `AsyncAfterCommit`—The scheduled path runs asynchronously after a save.

Field Name	Field Type	Description
`recordField`	string	Field used to determine when the scheduled path executes. The field's object is defined in FlowStart.
`timeSource`	FlowScheduledPathTimeSource (enumeration of type string)	Specify if a field or event is used to determine when the scheduled path executes. Possible values are: • `RecordField` • `RecordTriggerEvent`

FlowScreen

Screens capture information from users and display information to users. It extends FlowNode and inherits all its fields.

Field Name	Field Type	Description
`actions`	FlowScreenAction[]	An array of screen actions. This field is available in API version 59.0 and later.
`allowBack`	boolean	Indicates whether to show (`true`) or hide (`false`) the **Previous** button on the screen at runtime. When true, the **Previous** button appears only if the user visited a previous screen in the flow path and if `showFooter` for the screen is set to `true`. Set this field to false when revisiting the previous screen triggers an action that you don't want repeated, such as a credit card transaction. This field is available in API version 26.0 and later. Default: `true` You can set either `allowBack` or `allowFinish` to false, but not both.
`allowFinish`	boolean	Indicates whether to show (`true`) or hide (`false`) the **Finish** button on the screen at runtime. When `true`, the **Finish** button appears only if the screen element is the end of a flow path, and if `showFooter` for the screen is set to `true`. The default value is `true`. Set to `false` if user is required to go back to a previous screen to continue or complete the flow. For example, don't include a **Finish** button on a screen that tells the user to go back and make corrections on a previous screen. You can set `allowBack` or `allowFinish` to `false`, but not both. This field is available in API version 26.0 and later.

Field Name	Field Type	Description
`allowPause`	boolean	Indicates whether to show (`true`) or hide (`false`) the **Pause** button on the screen at runtime. The default value is `true`. A flow screen displays the **Pause** button if all these conditions are `true`. • **Let users pause flows** is enabled in the organization's process automation settings. • `allowPause` for the screen is set to `true`. • If the flow is embedded in a Visualforce page, the <flow:interview> component has its `showAllowPause` attribute set to `true`. • The `showFooter` field for the screen is set to `true`. This field is available in API version 33.0 and later.
`backButtonLabel`	string	A label for the Back button.
`connector`	FlowConnector	Specifies which node to execute after the screen node.
`fields`	FlowScreenField[]	An array of fields to display on the screen.
`helpText`	string	Text that appears if the end user clicks a link for help text. Supports merge fields in API version 26.0 and later.
`nextOrFinishButtonLabel`	string	A label for the Next or Finish button.
`pauseButtonLabel`	string	A label for the Pause button.
`pausedText`	string	A confirmation message that appears when an end user clicks **Pause**. This field is available in API version 33.0 and later.
`rules`		Reserved for future use.
`showFooter`	boolean	Indicates whether to show (`true`) or hide (`false`) the screen's footer at Lightning runtime. Classic runtime isn't supported. The default value is `true`. The footer includes navigation actions for the screen. If `showFooter` is hidden, use Lightning components on the screen to show navigation actions. This field is available in API version 42.0 and later.
`showHeader`	boolean	Indicates whether to show (`true`) or hide (`false`) the screen's header at Lightning runtime. Classic runtime isn't supported. The default value is `true`. The header includes access to help text for the screen. If `showHeader` is hidden, use Lightning components on the screen to show help text.

Field Name	Field Type	Description
		This field is available in API version 42.0 and later.
`stageReference`	FlowElementReferenceOrValue	The API name of the stage resource that's associated with the screen.
`triggers`	FlowScreenTrigger[]	An array of triggers configured for a flow screen field or a flow screen field attribute. This field is available in API version 59.0 and later.

FlowScreenAction

Defines an action that can be triggered by one or more flow screen components.

This metadata type is available in API version 59.0 and later.

Field Name	Field Type	Description
`actionName`	string	Required. The API name of the flow screen action.
`actionType`	InvocableActionType (enumeration of type string)	Required. The flow screen action type. Valid values are: • `flow`—Invokes an autolaunched flow.
`inputParameters`	FlowScreenActionInputParameter[]	An array of input parameters from the flow to the flow screen action.
`label`	string	Requiired. The label of the flow screen action.
`nameSegment`	string	The API name of the flow screen action.
`versionSegment`	int	The version number for the screen action.
`versionString`	string	Specifies the version of the screen action to be invoked. This field is available in API version 63.0 and later.

FlowScreenActionInputParameter

Defines an iput parameter for a flow screen action. It extends FlowScreenFieldInputParameter and inherits all its fields.

This metadata type is available in API version 59.0 and later.

Field Name	Field Type	Description
name	string	Required. Unique name of the input parameter of a flow screen action.

Field Name	Field Type	Description
value	FlowElementReferenceOrValue	Defines the valueof an input parameter for a flow screen action.

FlowScreenField

Represents a screen component. FlowScreenField extends FlowElement and inherits all its fields.

Field Name	Field Type	Description
`choiceReferences`	string[]	An array of references to FlowChoices or FlowDynamicChoiceSets. The resulting choice options appear in the order specified in this array, where the element at index 0 provides the top-most choice option. Supported for these types of screen components. • RadioButtons • DropdownBox • MultiSelectCheckboxes • MultiSelectPicklist Multi-select checkboxes and multi-select picklist fields are available in API version 26.0 and later.
`dataType`	FlowDataType (enumeration of type string)	Data type of the screen component. Only supported for the InputField, RadioButtons, and DropdownBox types of screen components. Valid data types are: • `Boolean` • `Currency` • `Date` • `DateTime` • `Number` • `String` Boolean input fields, which appear as checkbox fields at runtime, are available in API version 26.0 and later. Only the string data type is supported for multi-select checkboxes and multi-select picklist fields. Multi-select fields are available in API version 26.0 and later. Date/time input fields are available in API version 43.0 and later.

Field Name	Field Type	Description
`dataTypeMappings`	FlowDataTypeMapping[]	Reserved for future use.
`defaultSelectedChoiceReference`	string	The name of the FlowChoice element to use as the default value for the screen component. Supported for these types of screen components. • RadioButtons • DropdownBox • MultiSelectCheckboxes • MultiSelectPicklist For DropdownBox field types only, if `defaultSelectedChoiceReference` is empty or null, the reference at index 0 of `choiceReferences` is used as the default value. You can specify only one FlowChoice element as the default value for multi-select checkboxes and multi-select picklist fields. Multi-select fields are available in API version 26.0 and later.
`defaultValue`	FlowElementReferenceOrValue	The value that is used by default when the screen component requires users to provide input. Only supported for InputField, LargeTextArea, and PasswordField.
`extensionName`	string	The name of the Lightning component to display. This field is available in API version 42.0 and later.
`fields`	FlowScreenField[]	An array of columns to display in a section, or an array of fields to display in a column. This field is available in API version 49.0 and later.
`fieldText`	string	Field label that is displayed on the screen. Supports merge fields.
`fieldType`	FlowScreenFieldType (enumeration of type string)	Required. The type of field to display on a flow screen. Valid values are: • `DisplayText` • `InputField` • `LargeTextArea` • `PasswordField` • `RadioButtons` • `DropdownBox`

Field Name	Field Type	Description
		• `MultiSelectCheckboxes`—This value is available in API version 26.0 and later. • `MultiSelectPicklist`—This value is available in API version 26.0 and later. • `ComponentInstance`—This value is available in API version 42.0 and later. • `ComponentChoice` and `ComponentInput`—This value is available in API version 48.0 and later for the `Survey` `processType` value only. • `Region`— Specifies that a screen field in a section is a column. This value is available in API version 51.0 and later. • `RegionContainer`—Specifies that a screen field is a section. This value is available in API version 51.0 and later. • `ObjectProvided`—Specifies that a screen field is a field from a Salesforce object. This value is available in API version 51.0 and later. At runtime, each multi-select field stores its field value as a concatenation of the user-selected choice values, separated by semicolons. Any semicolons in the selected choice values are removed when added to the multi-select field value.
`helpText`	string	Text that appears if the end user clicks the help icon ( ) for the screen component. Supports merge fields in API version 26.0 and later.
`inputParameters`	FlowScreenFieldInputParameter[]	An array of input parameters. Supported only when `fieldType` is `ComponentInstance`. This field is available in API version 42.0 and later.
`inputsOnNextNavToAssocScrn`	FlowScreenFieldInputsRevisited (enumeration of type string)	Controls whether the flow remembers the input value if the user moves to any screen and then returns to the screen component. Valid values are: • `UseStoredValues`—Uses values from when the user last visited this screen.

Field Name	Field Type	Description
		• `ResetValues`—Refreshes inputs to incorporate changes elsewhere in the flow. The default value is `UseStoredValues`. This property applies to screen components in API version 51.0 and later and to record fields on flow screens in API version 57.0 and later.
`isRequired`	boolean	Indicates whether the user must select a choice or provide input. Not supported for DisplayText or boolean inputField.
`isVisible`	boolean	
`objectFieldReference`	string	Specifies the Salesforce object field for an ObjectProvided field.
`outputParameters`	FlowScreenFieldOutputParameter[]	An array of output parameters. Supported only when `fieldType` is `ComponentInstance` and when `storeOutputAutomatically` is `false`. This field is available in API version 42.0 and later.
`regionContainerType`	FlowRegionContainerType (enumeration of type string)	Stores information about a section component header. Possible values include: • `SectionWithHeader` • `SectionWithoutHeader` Available only when the component type is `Section`. This field is available in API version 55.0 and later.
`scale`	int	Controls the number of digits to the right of the decimal point up to 17 places. If you leave this field blank or set it to zero, only whole numbers appear when your flow runs. Available only when the data type is Number or Currency. Corresponds to the Decimal Places field in Flow Builder.
`sourceTemplateApiName`	string	The API name of the template specified by the provider. This field is available in API version 62.0 and later.
`sourceTemplateProviderType`	string	The API name of the source that provides the template. This field is available in API version 62.0 and later.

Field Name	Field Type	Description
`storeOutputAutomatically`	boolean	Indicates whether the screen component's output parameters are automatically available in the flow without creating any variables. When the value is `true`, you can reference an output parameter by specifying the `name` of the screen component and the output parameter, such as `Mailing_Address.City`. Supported only when `fieldType` is `ComponentInstance`. This field is available in API version 47.0 and later.
`styleProperties`	FlowScreenFieldStyleProperties	Specifies the style properties of a screen component. This field is available in API version 64.0 and later.
`validationRule`	FlowInputValidationRule	A rule that's used to validate the user input when the screen component is of type InputField, LargeTextArea, or PasswordField.
`visibilityRule`	FlowVisibilityRule	A condition-based rule that's used to render or hide the screen component. This field is available in API version 47.0 and later.

FlowScreenFieldInputParameter

Defines an input parameter from the flow to the extension. It extends FlowBaseElement and inherits all its fields. FlowScreenFieldInputParameter is available in API version 42.0.

Field Name	Field Type	Description
`name`	string	Required. Unique name for the input parameter.
`value`	FlowElementReferenceOrValue	Defines the value of the input parameter.

FlowScreenFieldOutputParameter

Defines an output parameter from the extension to the flow. It extends FlowBaseElement and inherits all its fields. FlowScreenFieldOutputParameter is available in API version 42.0.

Field Name	Field Type	Description
`assignToReference`	string	Required. Specifies the variable to which you want to assign the output parameter value.
`name`	string	Required. Unique name for the output parameter.

FlowScreenFieldStyleProperties

Defines how a screen component looks on a screen element at run time.

This metadata type is available in API version 64.0 and later.

Field Name	Field type	Description
`width`	FlowElementReferenceOrValue	The number of columns the width of the screen component fills up in a screen element's 12-column wide spatial grid. Valid values are numbers `1` through `12`.
`verticalAlignment`	FlowElementReferenceOrValue	The vertical alignment of the screen component. Valid values are `top`, `middle`, `bottom`.

FlowScreenTrigger

Defines an event handler for a flow screen component.

This metadata type is available in API version 59.0 and later.

Field Name	Field Type	Description
eventName	string	Required. The API name of the event from eventSource that the trigger listens for.
eventSource	string	Required. The screen field, screen field attribute, action, or action attribute where eventName takes place.
handlers	FlowScreenTriggerHandler[]	An array of flow screen handlers to conditionally run when the configured event is recevied.

FlowScreenTriggerHandler

Defines conditions for a flow screen trigger handler.

This metadata type is available in API version 59.0 and later.

Field Name	Field Type	Description
`conditionLogic`	string	Specifies logic for the conditions. Valid values are: • `and`—Evaluates to `true` only if all its conditions evaluate to true • `or`—Evaluates to `true` if any of its conditions evaluate to true • Advanced logic like `1 AND (2 OR 3)`—Evaluates to `true` if the first condition is `true` and either the second or third condition if `true`. When you use advanced logic, the string must consist of 1,000 or fewer characters. This field is available in API version 63.0 and later.
`conditions`	FlowCondition[]	An array of conditions that must be `true` to trigger the handler. This field is available in API version 63.0 and later.
screenActionName	string	Required. The API name of the FlowScreenAction to run when conditions are met.

FlowStage

A section of your flow that can be represented in the UI, such as with breadcrumbs. It extends FlowElement and inherits all its fields.

When an interview starts, any stages where `isActive` is `true` are added to the `$Flow.ActiveStages` global variable, which holds a collection of stages. Each stage's `stageOrder` determines the order they're added in. The stage with the lowest `stageOrder` is assigned to the `$Flow.CurrentStage` global variable.

Field Name	Field Type	Description
`isActive`	boolean	Indicates whether the stage is active by default.
`label`	string	A user-friendly label for this stage.
`stageOrder`	int	Indicates how the stage is ordered against other stages. The `stageOrder` value must be unique within the flow.

FlowStageStep

A step resource defines a step within a stage node. This metadata type is available in API version 53.0 and later.

Field Name	Field Type	Description
actionName	string	Required. Name of the flow associated with the step.
actionType	InvocableActionType (enumeration of type string)	Required. The type of the step. Valid values are: • createFieldGnrnPromptImplResp—This value is available in API version 62.0 and later. • exploreConversation—This value is available in API version 61.0 and later. • stepBackground • stepInteractive • stepMuleSoft
assignees	FlowStageStepAssignee	An array of users, groups, or queues that are assigned to complete the interactive step.
canAssigneeEdit	boolean	Reserved for future use.
entryActionInputParameters	FlowStageStepEntryActionInputParameter[]	An array of input parameters from the step to the evaluation flow that are used as an entry condition for the step.
entryActionName	string	The name of the evaluation flow used as an entry condition for the step.
entryActionOutputParameters	FlowStageStepEntryActionOutputParameter[]	An array of output parameters from the evaluation flow to the step used to determine if the step can be started.
entryActionType	InvocableActionType (enumeration of type string)	The type of the evaluation flow used as a custom entry condition for the step. Valid values are: • EvaluationFlow • exploreConversation—This value is available in API version 61.0 and later.
entryConditionLogic	string	Defines how the entry requirements for a step are evaluated. Valid values are: • And • Or • Custom logic, such as (1 AND (2 OR 3)) • Formula

Field Name	Field Type	Description
`entryConditions`	FlowCondition[]	An array of requirements that must be met to start the step.
`exitActionInputParameters`	FlowStageStepExitActionInputParameter[]	An array of input parameters from the step to the evaluation flow. These parameters specify an exit condition for the step.
`exitActionName`	string	The name of the step exit evaluation flow.
`exitActionOutputParameters`	FlowStageStepExitActionOutputParameter[]	An array of output parameters from the evaluation flow to the step. These parameters specify an exit condition for the step.
`exitActionType`	InvocableActionType (enumeration of type string)	The type of the evaluation flow used as a custom exit condition for the step. The only possible value are: • `EvaluationFlow` • `exploreConversation`—This value is available in API version 61.0 and later. .
`exitConditionLogic`	string	Defines how the exit requirements for an interactive step are evaluated. Valid values are: • `And` • `Or` • `Custom logic, such as (1 AND (2 OR 3))` • `Formula`
`exitConditions`	FlowCondition[]	An array of requirements to be met for exiting an interactive step.
`inputParameters`	FlowStageStepInputParameter[]	An array of input parameters from the step to its associated flow.
`label`	string	Required. The label for the step.
`outputParameters`	FlowStageStepOutputParameter[]	An array of output parameters from a flow to its associated step.
`requiresAsyncProcessing`	boolean	Not used in API version 63.0.
`shouldLock`	boolean	Reserved for future use.
`stepSubtype`	FlowElementSubtype (enumeration of type string)	Reserved for internal use.

FlowStageStepAssignee

An assignee associated with an Interactive step. Applicable only for interactive steps. This metadata type is available in API version 53.0 and later.

Field Name	Field Type	Description
`assignee`	FlowElementReferenceOrValue	Names of the user, group, or queue assigned to the interactive step.
`assigneeType`	FlowStageStepAssigneeType (enumeration of type string)	Required. The type of the assignee associated with the interactive step. Valid values are: • `Group` • `Queue` • `User` • `invalid`— This value is available in API version 61.0 and later.

FlowStageStepEntryActionInputParameter

Defines an input parameter from the step to its associated evaluation flow. It extends FlowBaseElement and inherits all its fields. This metadata type is available in API version 53.0 and later.

Field Name	Field Type	Description
`name`	string	Required. The unique name for the input parameter of the evaluation flow used by a step as an entry condition.
`value`	FlowElementReferenceOrValue	Defines the value of the input parameter of the evaluation flow used by a step as an entry condition.

FlowStageStepEntryActionOutputParameter

Defines an output parameter from an evaluation flow used to determine if the step meets entry criteria. It extends FlowBaseElement and inherits all its fields. This metadata type is available in API version 53.0 and later.

Field Name	Field Type	Description
`assignToReference`	string	Reserved for future use.
`name`	string	Required. A unique name for the output parameter of the evaluation flow used by a step as an entry condition. Valid values are: • `isOrchestrationConditionMet`

FlowStageStepExitActionInputParameter

Defines an input parameter from the stage or step to its associated evaluation flow. It extends FlowBaseElement and inherits all its fields. This metadata type is available in API version 53.0 and later.

Field Name	Field Type	Description
`name`	string	Required. A unique name for the input parameter of the evaluation flow used by a stage or step as an exit condition.
`value`	FlowElementReferenceOrValue	Defines the value of the input parameter of the evaluation flow used by a stage or step as an exit condition.

FlowStageStepExitActionOutputParameter

Defines an output parameter from an evaluation flow used to determine if the stage or step meets exit criteria. It extends FlowBaseElement and inherits all its fields. This metadata type is available in API version 53.0 and later.

Field Name	Field Type	Description
`assignToReference`	string	Reserved for future use.
`name`	string	Required. A unique name for the output parameter of the evaluation flow used by a stage or step as an exit condition. The only possible value is `isOrchestrationConditionMet`.

FlowStageStepInputParameter

Defines an input parameter from the step to the flow. It extends FlowBaseElement and inherits all its fields. This metadata type is available in API version 53.0 and later.

Field Name	Field Type	Description
`name`	string	Required. Unique name for the input parameter for a flow associated with the step.
`value`	FlowElementReferenceOrValue	Defines the value of the input parameter of the flow associated with a step.

FlowStageStepOutputParameter

Defines an output parameter from the step to the flow. It extends FlowBaseElement and inherits all its fields. This metadata type is available in API version 53.0 and later.

Field Name	Field Type	Description
`assignToReference`	string	Reserved for future use.
`name`	string	Required. Unique name for the output parameter for a flow associated with the step.

FlowStart

Represents the flow's Start element, which specifies how the flow starts. In an autolaunched flow, the Start element also defines when and how frequently to run the flow. To run the flow only for specific records, the Start element can define filter criteria.

FlowStart extends FlowNode and inherits all its fields except `name` and `label`. This metadata type is available in API version 47.0 and later.

Field Name	Field Type	Description
`capabilityTypes`	FlowCapability[]	An array of capabilities that can pass data with the flow. Only one capability is supported in API version 60.0 and later. This field is available in API version 60.0 and later.
`connector`	FlowConnector	Specifies which element to execute first.
`dataGraph`	string	The data graph associated with the flow. Reference fields from this data graph throughout the flow. This field is available in API version 61.0 and later.
`dataTypeMappings`	FlowDataTypeMapping[]	An array of data type mappings for input and output values that have the generic sObject data type. This field is available in API version 63.0 and later.
~~`doesRequireRecordChangedToMeetCriteria`~~	boolean	If set to `true`, conditions evaluate to `true` only if the record didn't meet the required conditions before the triggering update but now meets the conditions after the update. This field is available in API version 50.0 and later.
`entryType`	FlowEntryType (enumeration of type string)	Specifies when a unified individual can join a flow. Valid values are: • `AfterCompletion`—Unified individuals can join the flow only after they complete all previous flow runs of the same flow definition. • `Always`—Unified individuals can always join the flow. This field is available in API version 60.0 and later.
`eventName`	string	The name of the automation event that triggers the automation event-triggered flow. Valid values are: • `trgrOnSmsSubscription` • `trgrOnEmailSubscription` • `trgrOnOrderPlacement` • The API name of a form • The API name of an external service This field is available in API version 61.0 and later.
`eventType`	InvocableActionType (enumeration of type string)	The type of the automation event that triggers the automation event-triggered flow. Valid values are: • `exploreConversation`—This value is available in API version 61.0 and later.

Field Name	Field Type	Description
		• `externalEvent` • `processWebStoreUserRgstr` • `trgrOnEmailBounceEngagement` • `trgrOnEmailLinkClickEngagement` • `trgrOnEmailOpenEngagement` • `trgrOnEmailSubscription` • `trgrOnFormSubmission` • `trgrOnOrderPlacement` • `trgrOnSmsDeliveryFailureEngagement` • `trgrOnSmsLinkClickEngagement` • `trgrOnSmsResponseEngagement` • `trgrOnSmsSubscription` • `trgrOnWebCartAbandoned` • `trgrOnWhatsAppDeliveredEngagement` • `trgrOnWhatsAppDlvrFailureEngmt` • `trgrOnWhatsAppLinkClickEngmt` • `trgrOnWhatsAppReadEngagement` • `trgrOnWhatsAppResponseEngmt` • `trgrOnWhatsAppSubscription`
`filterFormula`	string	A formula that's used to filter what records execute the flow during a save. Available only in record-triggered flows. This field is available in API version 55.0 and later.
`filterLogic`	string	The filter logic that's applied to the filter condition requirements. To require all conditions, use `AND`. To require any conditions, use `OR`. For custom condition logic, enter the entire logic string, for example `1 AND 2 OR (3 AND 4)`. This field is available in API version 50.0 and later.
`filters`	FlowRecordFilter[]	An array of filters to apply when retrieving records from the database. For example, filter accounts to include only the records that haven't been updated in the last 4 weeks.
`flowRunAsUser`	string	Specifies who to run the flow as. Possible values are: • `TriggeringUser`—Run the flow as the user that triggered the flow. • `DefaultWorkflowUser`—Run the flow as the default workflow user. This field is available in API version 60.0 and later.
`form`	string	Required only for form-triggered flows. The content key value for the form used to trigger the flow. This field is available in API version 59.0 and later.

Field Name	Field Type	Description
`inputs`	FlowStartInputParameter[]	An array of inputs to the Start element.
`object`	string	The object whose records you want to retrieve from the database. A flow interview starts for each record that meets the filter conditions.
`publishSegment`	boolean	Indicates whether to republish the segment and update segment membership before the flow runs or on the segment's Data Cloud publish schedule. When the value is `true`, the segment is immediately republished before the flow runs, and ignores the segment's publish schedule. When the value is `false`, the segment is republished on the segment's Data Cloud publish schedule, but the segment isn't republished if the schedule is set to `Do not refresh`. The default value is `false`. This field is available in API version 60.0 and later.
`recordTriggerType`	RecordTriggerType (enumeration of type string)	Specifies what type of record changes can start the flow. Possible values are: • `Create`—When a record is created. • `Update`—When a record is updated. • `CreateAndUpdate`—When a record is created and updated. • `Delete`—When a record is deleted. This value is available in API version 50.0 and later. • `None`—For flows that aren't record-triggered flows. This value is available in API version 55.0 and later. Available only when `triggerType` is `RecordBeforeSave` or `DataCloudDataChange`. This field is available in API version 48.0 and later.
`schedule`	FlowSchedule	Required when `triggerType` is `Scheduled`. Specifies when and how frequently the flow runs.
`scheduledPaths`	FlowScheduledPath[]	Specifies the flow's scheduled paths. This field is available in API version 51.0 and later.
`segment`	string	The segment used to trigger the flow. This field is available in API version 56.0 and later.
`TimeZoneSidKey`	string	Reserved for future use.
`triggeringDataGraph`	string	The API name of the data graph that includes the data model object that triggers the automation event-triggered flow. This field is available in API version 63.0 and later.

Field Name	Field Type	Description
`triggeringDataModelObjectPath`	string	The Data Cloud path to the data model object that triggers the automation event-triggered flow. This field is available in API version 63.0 and later.
`triggerType`	FlowTriggerType (enumeration of type string)	Specifies what causes the flow to run. If you exclude this field, the flow has no trigger and starts only when a user or app launches the flow. Possible values are: • `AutomationEvent`—The flow starts when an automation event such as an SMS subscription occurs. This value is available in API version 62.0 and later. • `Capability`—When `capabilityTypes` is set, the flow starts when the capability is run. This value is available in API version 60.0 and later. • `DataCloudDataChange`— The flow starts when data model object (DMO) or calculated insight object (CIO) conditions are met. This value is available in API version 59.0 and later. • `DataGraphDataChange`— The flow starts when conditions are met in the specified data graph field. This value is available in API version 63.0 and later. • `EventDrivenJourney`—Reserved for internal use. • `ExternalSystemChange` • —The flow starts when a relevant change is detected in an external system. This value is available in API version 62.0 and later. • `PlatformEvent`—The flow starts when a platform event message is received. This value is available in API version 49.0 and later. • `RecordAfterSave`—The flow starts after a record is saved. This value is available in API version 49.0 and later. • `RecordBeforeDelete`—Deleting a record triggers an autolaunched flow before the record is deleted from the database. This value is available in API version 50.0 and later. • `RecordBeforeSave`—Creating and/or updating a record triggers an autolaunched flow to make more updates to that record before it's saved to the database. This value is available in API version 48.0 and later. • `Scheduled`—The flow starts at the scheduled time. This value is available in API version 47.0 and later. • `ScheduledJourney`— The flow starts only at the scheduled time and frequency. This value is available in API version 49.0 and later.

Field Name	Field Type	Description
		• `Segment`— At the scheduled time, the flow send emails to individuals included in the chosen segment. This value is available in API version 56.0 and later. Available only when `processType` is `AutoLaunchedFlow` or `PromptFlow`. This field is available in API version 47.0 and later.

FlowCapability

Defines the data structure of a capability. When the capability is invoked, it triggers the flow to run and data is passed between the flow and capability. It extends FlowElement and inherits all of its fields. This metadata type is available in API version 60.0 and later.

Field Name	Field Type	Description
`capabilityName`	string	Required. The specified capability that the flow integrates with. The valid format is *`Name`*`://`*`Name`*, for example, PromptBuilder://SalesEmail
`inputs`	FlowCapabilityInput[]	An array of capability inputs. The flow sets the input values and passes the data to the capability.

FlowCapabilityInput

Defines the data structure of a capability input. It extends FlowElement and inherits all of its fields. This metadata type is available in API version 60.0 and later.

Field Name	Field Type	Description
`capabilityInputName`	string	Required. The input name is the same for the capability and the flow.
`dataType`	string	The data type of the capability input. Valid types are: • `Boolean`—This value is available in API version 61.0 and later. • `Currency`—This value is available in API version 61.0 and later. • `Date`—This value is available in API version 61.0 and later. • `Number`—This value is available in API version 61.0 and later. • `sObject`—This value corresponds to a record variable. This value is available in API version 60.0 and later. • `String`—This value is available in API version 61.0 and later.
`isCollection`	boolean	Required. Indicates whether the input is a collection of values. The default value is `false`.

FlowStartInputParameter

Defines an input parameter to the flow Start element. It extends FlowBaseElement and inherits all its fields. This metadata type is available in API version 62.0 and later.

Field Name	Field Type	Description
`name`	string	Required. The unique name for the input parameter to the Start element.
`value`	FlowElementReferenceOrValue	Defines the value of the input parameter to the Start element.

FlowStep

Steps function as placeholders when you're building a flow. It extends FlowNode and inherits all its fields.

Field Name	Field Type	Description
`connectors`	FlowConnector[]	Specifies which node to execute after the step node.

FlowSubflow

A subflow element references another flow, which it calls at run time. The flow that contains the subflow element is referred to as the parent flow. FlowSubflow extends FlowNode and inherits all its fields. It's available in API version 25.0 and later.

Field Name	Field Type	Description
`connector`	FlowConnector	Specifies which node to execute after the subflow.
`flowName`	string	References the flow to call at runtime. The value must be an API name of a flow and it can't contain an appended hyphen and version number.
`inputAssignments`	FlowSubflowInputAssignment[]	An array of input variable assignments that are set at the start of the flow.
`outputAssignments`	FlowSubflowOutputAssignment[]	An array of output variable assignments that are set at the end of the flow.
`storeOutputAutomatically`	boolean	Indicates whether the subflow's output parameters are automatically available in the flow without creating any variables. When the value is `true`, you can reference an output parameter by specifying the API name of the subflow in the flow. When the value is `false`, create variables manually to store output values from the subflow. The default value is `false`. This field is available in API version 49.0 and later.

FlowSubflowInputAssignment

Assigns an element or value from the parent flow to a variable in the referenced flow. Input assignments occur when the subflow calls the referenced flow. It extends FlowBaseElement and inherits all its fields. It's available in API version 25.0 and later.

Field Name	Field Type	Description
`name`	string	Required. Unique name for the variable in the referenced flow.
`value`	FlowElementReferenceOrValue	Defines the value to assign to the variable.

FlowSubflowOutputAssignment

Assigns the value of a variable from the referenced flow to a variable in the parent flow. Output assignments occur when the referenced flow is finished running. It extends FlowBaseElement and inherits all its fields. It's available in API version 25.0 and later.

Field Name	Field Type	Description
`assignToReference`	string	Unique name for the variable in the parent flow.
`name`	string	Required. Unique name for the variable in the referenced flow.

FlowTransform

Defines a node that can dynamically transform the value of source data to target data in the flow. It extends FlowNode and inherits all of its fields. This metadata type is available in API version 59.0 and later.

Field Name	Field Type	Description
`apexClass`	string	The Apex class of the target data after transformation if its data type is `Apex`.
`connector`	FlowConnector[]	Specifies which node to execute after this data transformation.
`dataType`	FlowDataType (enumeration of type string)	Required. Specifies the data type of the transformed data. In Flow Builder, it corresponds to the target data in the Transform element. Valid types are: • `Apex` • `Boolean`—This value is available in API version 62.0. • `Currency`—This value is available in API version 62.0. • `Date`—This value is available in API version 62.0. • `DateTime`—This value is available in API version 62.0. • `Number`—This value is available in API version 62.0. • `String`—This value is available in API version 62.0. • `sObject`—This value corresponds to a record variable.

Field Name	Field Type	Description
`isCollection`	boolean	Indicates whether the variable is a collection of values. The default value is `false`.
`objectType`	string	Object type of this variable resource if its data type is `sObject`.
`scale`	int	Controls the number of digits to the right of the decimal point up to 17 places. If you leave this field blank or set it to zero, only whole numbers appear when your flow runs. Corresponds to the Decimal Places field in Flow Builder.
`storeOutputAutomatically`	boolean	Reserved for future use.
`transformValues`	FlowTransformValue[]	An array of values for data transformation

FlowTransformValue

Defines the values for transforming specific data in the flow. It extends FlowBaseElement and inherits all its fields. This metadata type is available in API version 59.0 and later.

Field Name	Field Type	Description
`transformValueActions`	FlowTransformValueAction[]	An array of actions for data transformation
`transformValueName`	string	Reserved for future use.
`transformValueLabel`	string	Reserved for future use.
`transformValueDescription`	string	Reserved for future use.

FlowTransformValueAction

Defines the data and actions to transform in the flow. It extends FlowBaseElement and inherits all its fields. This metadata type is available in API version 59.0 and later.

Field Name	Field Type	Description
`actionName`	string	Reserved for future use.
`actionType`	InvocableActionType (enumeration of type string)	Reserved for future use.
`actionVersionString`	string	Reserved for future use.
`assignToReference`	string	Reserved for future use.
`inputParameters`	FlowTransformValueActionInputParameter[]	An array of input parameters for data transformation. This field is available in API version 60.0 and later.
`outputFieldApiName`	string	The API name of the field for transformed data in a data transformation mapping. In Flow Builder, it corresponds to the target data field in the Transform element.

Field Name	Field Type	Description
`transformType`	FlowTransformValueActionType (enumeration of type string)	Required. The type of transformation from source data to target data. Valid types are: • `Count`—Calculates the number of items in a source collection. • `GetItemByIndex`—Reserved for future use. • `InnerJoin`—Joins selected data from two source collections that are stored in a target collection in a flow. This value is available in API version 63.0 and later. See `complexValueType` on FlowElementReferenceOrValue. • `InvocableAction`—Reserved for future use. • `Map`—Specifies a mapping between the datasets in flows. In Flow Builder, it corresponds to the mapping between source data fields and target data fields. • `Sum`—Adds the numeric values of a field on each item in a collection.
`value`	FlowElementReferenceOrValue	Defines the value of the transformed data. In Flow Builder, the value of this field corresponds to the result of the target data field in the Transform element.

FlowTransformValueActionInputParameter

Defines the input parameters of the source data for data transformation. This metadata type is available in API version 60.0 and later.

Field Name	Field Type	Description
`name`	string	A key that specifies the configuration of input parameters for this data transformation when `transformType` is set to `Sum` or `Count`. Valid values are: • `aggregationField`—The field on each item in a source collection that's used to calculate the transformed value. • `aggregationValues`—The source collection that's used to calculate the transformed value.
`value`	FlowElementReferenceOrValue	Defines the value of the specified key in `name`.

FlowTextTemplate

Defines a text template that can be used throughout the flow. It extends FlowElement and inherits all its fields.

Field Name	Field Type	Description
`isViewedAsPlainText`	boolean	If set to `true`, the flow resource remembers the View as Plain Text setting used for the text template after the flow resource is saved.

Field Name	Field Type	Description
		If set to `false`, the flow resource uses the View as Rich Text setting. The default value is `false`.
`text`	string	Actual text of the template. Supports merge fields.

FlowVariable

With variables, creates updatable values to use in the flow. FlowVariable extends FlowElement and inherits all its fields.

Field Name	Field Type	Description
`apexClass`	string	The Apex class of this variable if its data type is `Apex`. This field is available in API version 46.0 and later.
`dataType`	FlowDataType (enumeration of type string)	Required. Valid types are: • `Apex`—This value is available in API version 46.0 and later. • `Boolean` • `Currency` • `Date` • `DateTime`—This value is available in API version 30.0 and later. • `Number` • `Multipicklist`—This value is available in API version 34.0 and later. • `Picklist`—This value is available in API version 34.0 and later. • `String` • `sObject`—This value corresponds to a record variable.
`isCollection`	boolean	Indicates whether the variable is a collection of values. This field is available in API version 30.0 and later. In API version 32.0 and later, a collection variable can be of any data type. The default value is `False`.
`isInput`	boolean	Indicates whether the variable can be set at the start of the flow using URL parameters, Visualforce controllers, or subflow inputs. This field is available in API version 25.0 and later.

Field Name	Field Type	Description
		Default value: • `False` for a variable created in API version 25.0 and later or in the Flow Builder in Summer '12 and later. • `True` for a variable created in API version 24.0 or in Flow Builder in Spring '12 and earlier. Disabling input or output access for an existing variable can break the functionality of applications and pages that call the flow and access the variable. For example, you can access variables from URL parameters, processes, and other flows.
`isOutput`	boolean	Indicates whether the variable's value can be accessed from Visualforce controllers and other flows. This field is available in API version 25.0 and later. Default value: • `False` for a variable created in API version 25.0 and later or in the Flow Builder in Summer '12 and later. • `True` for a variable created in API version 24.0 or in Flow Builder in Spring '12 and earlier. Disabling input or output access for an existing variable can break the functionality of applications and pages that call the flow and access the variable. For example, you can access variables from URL parameters, processes, and other flows.
`objectType`	string	Object type of this variable if its data type is `sObject`.
`scale`	int	Controls the number of digits to the right of the decimal point up to 17 places. If you leave this field blank or set it to zero, only whole numbers appear when your flow runs. Corresponds to the Decimal Places field in Flow Builder.

Field Name	Field Type	Description
`value`	FlowElementReferenceOrValue	Default value of this variable. Default values aren't supported if the variable's data type is `Picklist` or `Multipicklist`.

FlowVisibilityRule

Visibility rules render a flow screen component when visibility rule conditions are met. Hides a flow screen component when visibility rule conditions aren't met. This metadata type is available in API version 47.0 and later.

Field Name	Field Type	Description
`conditionLogic`	string	Specifies logic for the conditions. Value can be: • `and`—Evaluates to `true` only if all its conditions evaluate to true. • `or`—Evaluates to `true` if any of its conditions evaluate to true. • Advanced logic like `1 AND (2 OR 3)`—Evaluates to `true` if the first condition is `true` and either the second or third condition is `true`. When you use advanced logic, the string must consist of 1,000 or fewer characters.
`conditions`	FlowCondition[]	An array of conditions that must be true for the flow to wait for this event.

FlowWait

Waits for one or more defined events to occur. FlowWait extends FlowNode and inherits all its fields. FlowWait is available in API version 32.0 and later.

Field Name	Field Type	Description
`defaultConnector`	FlowConnector	Specifies which node to execute if the conditions are false for every event in the Wait element.
`defaultConnectorLabel`	string	Label for the default connector.
`faultConnector`	FlowConnector	Specifies which node to execute if the attempt to wait results in an error. If any of the wait events fail, the flow takes the fault connector.
`timeZoneId`	string	Reserved for future use.

Field Name	Field Type	Description
`waitEvents`	FlowWaitEvent[]	An array of events that the Wait element is waiting for. If the conditions for every event evaluate to `false`, the `defaultConnector` is used.

FlowWaitEvent

An event that a FlowWait element is waiting for. FlowWaitEvent extends FlowElement and inherits all its fields. FlowWaitEvent is available in API version 32.0 and later.

Field Name	Field Type	Description
`conditionLogic`	string	Specifies logic for the conditions. Value can be: • `and`—Evaluates to `true` only if all its conditions evaluate to `true` • `or`—Evaluates to `true` if any of its conditions evaluate to `true` • Advanced logic like `1 AND (2 OR 3)`—Evaluates to `true` if the first condition is `true` and either the second or third condition is `true` When you use advanced logic, the string must consist of 1,000 or fewer characters.
`associatedElement`	string	The API name of the event that resumes the flow. This field is available in API version 60.0 and later.
`conditions`	FlowCondition[]	An array of conditions that must be `true` for the flow to wait for this event.
`automationEventName`	string	Name of the automation event that the Wait element is waiting for.
`automationEventType`	InvocableActionType (enumeration of type string)	The type of the automation event that triggers the Wait element is waiting for. Valid values are: • `exploreConversation`—This value is available in API version 61.0 and later. • `trgrOnSmsSubscription` • `trgrOnEmailSubscription`

Field Name	Field Type	Description
		• `trgrOnOrderPlacement` • `trgrOnFormSubmission`
`connector`	FlowConnector	Specifies which node to execute if this event is the first event that occurs.
`eventType`	string	Required. The event's type. The type determines which input parameters are available to define this event. Valid values are: • `AlarmEvent`—This event is an alarm based off an absolute date/time value. • `DateRefAlarmEvent`—This event is an alarm based off a date/time field on a record.
`extendUntil`	Time	Reserved for future use.
`filters`	FlowRecordFilter[]	An array of filters to apply when retrieving records from the database. For example, filter accounts to include only the records that haven't been updated in the last 4 weeks. This field is available in API version 60.0 and later.
`filterlogic`	string	The filter logic that's applied to the filter condition requirements. To require all conditions, use AND. To require any conditions, use OR. For custom condition logic, enter the entire logic string, for example 1 AND 2 OR (3 AND 4). This field is available in API version 60.0 and later.
`inputParameters`	FlowWaitEventInputParameter[]	An array of the event's input parameters. The parameter values are set by using values from the flow.
`interactionType`	FlowWaitInteractionType (enumeration of type string)	Specifies what type of event can resume the flow. Possible values are: • `SmsResponse`—An SMS response event • `WhatsappResponse`—A WhatsApp response event This field is available in API version 62.0 and later.
`label`	string	Required. Label for the wait event.

Field Name	Field Type	Description
`object`	string	The object that contains the event you want to use to resume the flow. This field is available in API version 60.0 and later.
`offset`	int	Reserved for future use.
`offsetUnit`	FlowScheduledPathOffsetUnit (enumeration of type string)	Reserved for future use.
`outputParameters`	FlowWaitEventOutputParameter[]	An array of the event's output parameters. The parameter values are assigned from the event to variables in the flow.
`recordTriggerType`	RecordTriggerType	Specifies what type of record changes can resume the flow. Possible values are: • `Create`—When a related record is created • `Update`—When a related record is updated • `CreateAndUpdate`—When a related record is created and updated This field is available in API version 60.0 and later.

FlowWaitEventInputParameter

An input parameter for FlowWaitEvent. The parameter's value is set by using values from the flow. It extends FlowBaseElement and inherits all its fields. FlowWaitEventInputParameter is available in API version 32.0 and later.

Field Name	Field Type	Description
`name`	string	Unique name for the input parameter.
`value`	FlowElementReferenceOrValue	Defines the value of the input parameter.

FlowWaitEventOutputParameter

An output parameter for FlowWaitEvent. The parameter's value is assigned to a variable in the flow so that it can be referenced in another part of the flow. It extends FlowBaseElement and inherits all its fields. FlowWaitEventOutputParameter is available in API version 32.0 and later.

Field Name	Field Type	Description
`assignToReference`	string	Required. Specifies the variable to which you want to assign the output parameter value.

Field Name	Field Type	Description
`name`	string	Required. Unique name for the output parameter.

Upgrade Flow Files to API Version 44.0 or Later

In API version 43.0 and earlier, the Flow object's fullName field included the flow's version number. Starting in API version 44, the field no longer includes the version number. Before you deploy using API version 44.0 via Metadata API or Salesforce CLI, make sure that:

- The `flows` directory doesn't include any unused flow versions.
- For each active flow, the `status` field is `Active`. Any flow without a `status` value is deployed or retrieved with a `status` value of `Draft`.
- The `flowDefinitions` directory is empty.

For Metadata API only.

- The `package.xml` file is set to API version 44.0.
- For the latest version of each flow, the file name doesn't include a version number. For example, change `myflow-3.flow` to `myflow.flow`.

For Salesforce CLI only.

- The `sfdx-project.json` file is set to `"sourceApiVersion": "44.0"`.
- For the latest version of each flow, the file name doesn't include a version number. For example, change `myflow-1.flow-meta.xml` to `myflow.flow-meta.xml`.

As part of this upgrade, flow definitions are no longer necessary when you deploy or retrieve via Metadata API. If you deploy with flow definitions, the active version numbers in the flow definitions override the **status** fields in the flows. For example, the active version number in the flow definition is version 3, and the latest version of the flow is version 4 with the **status** field as `Active`. After you deploy your flow, the active version is version 3.

After you finished this upgrade, you can integrate with a version control system without worrying about flow file names changing. To reduce deployment issues when you push the source code into a scratch org, make sure that you don't reuse an existing scratch org.

For more information, see Deploy Processes and Flows as Active in *Salesforce Help*.

Declarative Metadata Sample Definition

Here's a sample XML definition of a flow.

```
<?xml version="1.0" encoding="UTF-8"?>
<Flow xmlns="http://soap.sforce.com/2006/04/metadata">
    <actionCalls>
        <name>Get_Info</name>
        <label>Get Info</label>
        <locationX>380</locationX>
        <locationY>242</locationY>
        <actionName>GetFirstFromCollection</actionName>
        <actionType>apex</actionType>
        <connector>
            <targetReference>Update_If_Existing</targetReference>
        </connector>
        <dataTypeMappings>
```

```
        <typeName>T__inputCollection</typeName>
        <typeValue>Account</typeValue>
    </dataTypeMappings>
    <dataTypeMappings>
        <typeName>U__outputMember</typeName>
        <typeValue>Account</typeValue>
    </dataTypeMappings>
    <flowTransactionModel>CurrentTransaction</flowTransactionModel>
    <inputParameters>
        <name>inputCollection</name>
        <value>
            <elementReference>accts.accounts</elementReference>
        </value>
    </inputParameters>
    <nameSegment>GetFirstFromCollection</nameSegment>
    <storeOutputAutomatically>true</storeOutputAutomatically>
    <versionSegment>1</versionSegment>
</actionCalls>
<actionCalls>
    <name>Post_to_Contact_s_Feed</name>
    <label>Post to Contact&amp;apos;s Feed</label>
    <locationX>50</locationX>
    <locationY>890</locationY>
    <actionName>chatterPost</actionName>
    <actionType>chatterPost</actionType>
    <connector>
        <targetReference>Confirm</targetReference>
    </connector>
    <flowTransactionModel>CurrentTransaction</flowTransactionModel>
    <inputParameters>
        <name>text</name>
        <value>
            <elementReference>chatterMessage</elementReference>
        </value>
    </inputParameters>
    <inputParameters>
        <name>subjectNameOrId</name>
        <value>
            <elementReference>contact.Id</elementReference>
        </value>
    </inputParameters>
    <nameSegment>chatterPost</nameSegment>
    <storeOutputAutomatically>true</storeOutputAutomatically>
    <versionSegment>1</versionSegment>
</actionCalls>
<apiVersion>49.0</apiVersion>
<assignments>
    <name>Set_Contact_ID</name>
    <label>Set Contact ID</label>
    <locationX>50</locationX>
    <locationY>674</locationY>
    <assignmentItems>
        <assignToReference>contact.Id</assignToReference>
        <operator>Assign</operator>
```

```
            <value>
                <elementReference>existingId</elementReference>
            </value>
        </assignmentItems>
        <connector>
            <targetReference>Update_Contact</targetReference>
        </connector>
    </assignments>
    <decisions>
        <name>Update_If_Existing</name>
        <label>Update If Existing?</label>
        <locationX>380</locationX>
        <locationY>350</locationY>
        <defaultConnector>
            <isGoTo>true</isGoTo>
            <targetReference>Create_Contact</targetReference>
        </defaultConnector>
        <defaultConnectorLabel>No</defaultConnectorLabel>
        <rules>
            <name>Update_Yes</name>
            <conditionLogic>and</conditionLogic>
            <conditions>
                <leftValueReference>updateExisting</leftValueReference>
                <operator>EqualTo</operator>
                <rightValue>
                    <booleanValue>true</booleanValue>
                </rightValue>
            </conditions>
            <connector>
                <targetReference>Find_a_Match</targetReference>
            </connector>
            <label>Yes</label>
        </rules>
    </decisions>
    <decisions>
        <name>Update_or_Create</name>
        <label>Update or Create?</label>
        <locationX>182</locationX>
        <locationY>566</locationY>
        <defaultConnector>
            <targetReference>Create_Contact</targetReference>
        </defaultConnector>
        <defaultConnectorLabel>Create New</defaultConnectorLabel>
        <rules>
            <name>Update_Existing</name>
            <conditionLogic>and</conditionLogic>
            <conditions>
                <leftValueReference>existingId</leftValueReference>
                <operator>IsNull</operator>
                <rightValue>
                    <booleanValue>false</booleanValue>
                </rightValue>
            </conditions>
            <connector>
```

```
                <targetReference>Set_Contact_ID</targetReference>
            </connector>
            <label>Update Existing</label>
        </rules>
    </decisions>
    <dynamicChoiceSets>
        <name>accounts</name>
        <dataType>String</dataType>
        <displayField>Name</displayField>
        <object>Account</object>
        <outputAssignments>
            <assignToReference>contact.AccountId</assignToReference>
            <field>Id</field>
        </outputAssignments>
        <valueField>Id</valueField>
    </dynamicChoiceSets>
    <environments>Default</environments>
    <formulas>
        <name>created_or_updated</name>
        <dataType>String</dataType>
        <expression>IF({!Create_Contact}, &quot;created&quot;,
&quot;updated&quot;)</expression>
    </formulas>
    <interviewLabel>New Contact {!$Flow.CurrentDateTime}</interviewLabel>
    <isAdditionalPermissionRequiredToRun>true</isAdditionalPermissionRequiredToRun>
    <isTemplate>true</isTemplate>
    <label>New Contact</label>
    <processMetadataValues>
        <name>BuilderType</name>
        <value>
            <stringValue>LightningFlowBuilder</stringValue>
        </value>
    </processMetadataValues>
    <processMetadataValues>
        <name>CanvasMode</name>
        <value>
            <stringValue>AUTO_LAYOUT_CANVAS</stringValue>
        </value>
    </processMetadataValues>
    <processMetadataValues>
        <name>OriginBuilderType</name>
        <value>
            <stringValue>LightningFlowBuilder</stringValue>
        </value>
    </processMetadataValues>
    <processType>Flow</processType>
    <recordCreates>
        <name>Create_Contact</name>
        <label>Create Contact</label>
        <locationX>314</locationX>
        <locationY>674</locationY>
        <connector>
            <isGoTo>true</isGoTo>
            <targetReference>Post_to_Contact_s_Feed</targetReference>
```

```
        </connector>
        <inputReference>contact</inputReference>
    </recordCreates>
    <recordLookups>
        <name>Find_a_Match</name>
        <label>Find a Match</label>
        <locationX>182</locationX>
        <locationY>458</locationY>
        <assignNullValuesIfNoRecordsFound>true</assignNullValuesIfNoRecordsFound>
        <connector>
            <targetReference>Update_or_Create</targetReference>
        </connector>
        <filterLogic>and</filterLogic>
        <filters>
            <field>FirstName</field>
            <operator>EqualTo</operator>
            <value>
                <elementReference>contact.FirstName</elementReference>
            </value>
        </filters>
        <filters>
            <field>LastName</field>
            <operator>EqualTo</operator>
            <value>
                <elementReference>contact.LastName</elementReference>
            </value>
        </filters>
        <object>Contact</object>
        <outputAssignments>
            <assignToReference>existingId</assignToReference>
            <field>Id</field>
        </outputAssignments>
    </recordLookups>
    <recordUpdates>
        <name>Update_Contact</name>
        <label>Update Contact</label>
        <locationX>50</locationX>
        <locationY>782</locationY>
        <connector>
            <targetReference>Post_to_Contact_s_Feed</targetReference>
        </connector>
        <inputReference>contact</inputReference>
    </recordUpdates>
    <screens>
        <name>Confirm</name>
        <label>Confirm</label>
        <locationX>50</locationX>
        <locationY>998</locationY>
        <allowBack>false</allowBack>
        <allowFinish>true</allowFinish>
        <allowPause>true</allowPause>
        <fields>
            <name>confirmation_message</name>
          <fieldText>Thanks! &lt;a href=&quot;/{!contact.Id}&quot;&gt;The contact&lt;/a&gt;
```

```
 was {!created_or_updated}.</fieldText>
            <fieldType>DisplayText</fieldType>
        </fields>
        <showFooter>true</showFooter>
        <showHeader>true</showHeader>
    </screens>
    <screens>
        <name>Contact_Info</name>
        <label>Contact Info</label>
        <locationX>380</locationX>
        <locationY>134</locationY>
        <allowBack>true</allowBack>
        <allowFinish>true</allowFinish>
        <allowPause>true</allowPause>
        <connector>
            <targetReference>Get_Info</targetReference>
        </connector>
        <fields>
            <name>contactName</name>
            <extensionName>flowruntime:name</extensionName>
            <fieldType>ComponentInstance</fieldType>
            <inputsOnNextNavToAssocScrn>UseStoredValues</inputsOnNextNavToAssocScrn>
            <isRequired>true</isRequired>
            <outputParameters>
                <assignToReference>contact.FirstName</assignToReference>
                <name>firstName</name>
            </outputParameters>
            <outputParameters>
                <assignToReference>contact.LastName</assignToReference>
                <name>lastName</name>
            </outputParameters>
        </fields>
        <fields>
            <name>Account</name>
            <choiceReferences>accounts</choiceReferences>
            <dataType>String</dataType>
            <fieldText>Account</fieldText>
            <fieldType>DropdownBox</fieldType>
            <isRequired>true</isRequired>
        </fields>
        <fields>
            <name>update_toggle</name>
            <extensionName>flowruntime:toggle</extensionName>
            <fieldType>ComponentInstance</fieldType>
            <inputParameters>
                <name>label</name>
                <value>
                    <stringValue>If this contact already exists, update the existing
record.</stringValue>
                </value>
            </inputParameters>
            <inputParameters>
                <name>messageToggleActive</name>
                <value>
```

```
                <stringValue>Update existing</stringValue>
            </value>
        </inputParameters>
        <inputParameters>
            <name>messageToggleInactive</name>
            <value>
                <stringValue>Create other contact</stringValue>
            </value>
        </inputParameters>
        <inputsOnNextNavToAssocScrn>UseStoredValues</inputsOnNextNavToAssocScrn>
        <isRequired>true</isRequired>
        <outputParameters>
            <assignToReference>updateExisting</assignToReference>
            <name>value</name>
        </outputParameters>
    </fields>
    <showFooter>true</showFooter>
    <showHeader>true</showHeader>
</screens>
<start>
    <locationX>254</locationX>
    <locationY>0</locationY>
    <connector>
        <targetReference>Contact_Info</targetReference>
    </connector>
</start>
<status>Draft</status>
<textTemplates>
    <name>chatterMessage</name>
    <isViewedAsPlainText>false</isViewedAsPlainText>
    <text>The contact was {!created_or_updated}.</text>
</textTemplates>
<variables>
    <name>accts</name>
    <apexClass>ComplexObjectExample</apexClass>
    <dataType>Apex</dataType>
    <isCollection>false</isCollection>
    <isInput>false</isInput>
    <isOutput>false</isOutput>
</variables>
<variables>
    <name>contact</name>
    <dataType>SObject</dataType>
    <isCollection>false</isCollection>
    <isInput>false</isInput>
    <isOutput>false</isOutput>
    <objectType>Contact</objectType>
</variables>
<variables>
    <name>existingId</name>
    <dataType>String</dataType>
    <isCollection>false</isCollection>
    <isInput>false</isInput>
    <isOutput>false</isOutput>
```

```
    </variables>
    <variables>
        <name>updateExisting</name>
        <dataType>Boolean</dataType>
        <isCollection>false</isCollection>
        <isInput>false</isInput>
        <isOutput>false</isOutput>
    </variables>
</Flow>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SEE ALSO:

Salesforce Help: Deploy Processes and Flows as Active

FlowCategory

Represents a list of flows that are grouped by category. Flows aren't added directly to a Lightning Bolt Solution. Instead, add the category the flows are in to the Lightning Bolt Solution. This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

FlowCategory components have the suffix `.flowCategory` and are stored in the `flowCategories` folder.

Version

FlowCategory components are available in API version 43.0 and later.

Fields

Field Name	Field Type	Description
`description`	string	The description of this flow category.
`flowCategoryItems`	FlowCategoryItems[]	The list of flows in this flow category.
`masterLabel`	string	Required. The label for this flow category, which appears in Setup.

FlowCategoryItems

Represents the list of flows in a flow category.

Field Name	Field Type	Description
flow	string	Required. The name of the flow.

Declarative Metadata Sample Definition

The following is an example of a FlowCategory component.

```
<?xml version="1.0" encoding="UTF-8"?>
<FlowCategory xmlns="http://soap.sforce.com/2006/04/metadata"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <flowCategoryItems>
        <flow>PausableFlow</flow>
    </flowCategoryItems>
    <flowCategoryItems>
        <flow>BankingFlow</flow>
    </flowCategoryItems>
    <masterLabel>updateBenefits</masterLabel>
        <description>All the update benefits.</description>
</FlowCategory>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>updateBenefits</members>
        <name>FlowCategory</name>
    </types>
    <version>43.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

FlowDefinition

Represents the flow definition's description and active flow version number.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Important: In API version 44.0, we recommend upgrading your flows to flow metadata file names without version numbers and discontinue using the FlowDefinition object to activate or deactivate a flow. Then use the Flow object to activate or deactivate a flow. For more information, see Upgrade Flow Files to API Version 44.0.

If you deploy with flow definitions, the active version numbers in the flow definitions override the `status` fields in the flows. For example, the active version number in the flow definition is version 3, and the latest version of the flow is version 4 with the `status` field as `Active`. After you deploy your flow, the active version is version 3.

Declarative Metadata File Suffix and Directory Location

FlowDefinitions are stored in the `flowDefinitions` directory of the corresponding package directory. The file name matches the flow definition's unique full name, and the extension is `.flowDefinition`.

Version

FlowDefinition is available in API version 34.0 and later.

Field Name	Field Type	Description
`activeVersionNumber`	int	The version number of the active flow.
`apiVersion`	int	Reserved for internal use.
`description`	string	Description of the flow definition.
`masterLabel`	string	Label for the flow definition. In managed packages, this field inherits the flow's active version name. To change this label from a subscriber's org, edit the packaged flow name.

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

FlowTest

Represents the metadata associated with a flow test. Before you activate a record-triggered flow, you can test it to verify its expected results and identify flow run-time failures.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

FlowTest components have the suffix `.flowtest` and are stored in the `flowtests` folder.

Version

FlowTest components are available in API version 55.0 and later.

Special Access Rules

There are no additional access requirements that are specific to this type.

Fields

Field Name	Description
`description`	**Field Type** string **Description** The description of the flow test, such as what it does or how it works.
`flowApiName`	**Field Type** string **Description** Required. The API name of the flow associated with the flow test.
`label`	**Field Type** string **Description** Required. The label of the flow test.
`testPoints`	**Field Type** FlowTestPoint[] **Description** An array of test points for the test.

FlowTestPoint

Defines a flow test point that is evaluated when a flow test runs. Each test point is evaluated in the order that it's listed.

Field Name	Description
`assertions`	**Field Type** FlowTestAssertion[] **Description** An array of assertions for the test.
`elementApiName`	**Field Type** string **Description** Required. The element API names for the start of the flow and the end of the flow.

Field Name	Description
	Possible values are: • `Start` • `Finish`
`parameters`	**Field Type** FlowTestParameter[] **Description** An array of parameters for the test.

FlowTestAssertion

Defines an assertion for a test point that is evaluated when a flow test runs. If one assertion evaluates to false, the test run fails.

Field Name	Description
`conditions`	**Field Type** FlowTestCondition[] **Description** An array of conditions for an assertion.
`errorMessage`	**Field Type** string **Description** The custom message that appears in Flow Builder if its associated condition evaluates to false.

FlowTestCondition

Defines a condition for an assertion that is evaluated when a flow test runs. If one condition evaluates to false, the assertion fails.

Field Name	Description
`leftValueReference`	**Field Type** string **Description** Required. The reference to the flow resource that the specified operator applies to.
`operator`	**Field Type** FlowComparisonOperator (enumeration of type string)

Field Name	Description
	Description Required. The operation that is applied to the resource reference in the `leftValueReference` field. Possible values are: • `Contains` • `EndsWith` • `EqualTo` • `GreaterThan` • `GreaterThanOrEqualTo` • `HasError`—This value is available in API version 64.0 and later. • `In`—This value is available in API version 56.0 and later. • `IsBlank`—This value is available in API version 61.0 and later. • `IsChanged` • `IsEmpty`—This value is available in API version 61.0 and later. • `IsNull` • `LessThan` • `LessThanOrEqualTo` • `NotEqualTo` • `NotIn`—This value is available in API version 56.0 and later. • `StartsWith` • `WasSelected` • `WasSet` • `WasVisited`
`rightValue`	**Field Type** FlowTestReferenceOrValue on page 1212 **Description** The value that the operator applies to the resource reference in the `leftValueReference` field.

FlowTestReferenceOrValue

Defines a specific value that the operator applies to the resource reference in flow test assertions and conditions.

Field Name	Description
`booleanValue`	**Field Type** boolean

Field Name	Description
	Description Specifies a boolean value.
`dateTimeValue`	**Field Type** dateTime **Description** Specifies a dateTime value.
`dateValue`	**Field Type** date **Description** Specifies a dateValue value.
`numberValue`	**Field Type** double **Description** Specifies a number value.
`sobjectValue`	**Field Type** string **Description** Specifies an sObject value.
`stringValue`	**Field Type** string **Description** Specifies a string value.

FlowTestParameter

Defines parameters for the triggering record in the Start test point.

Field Name	Description
`leftValueReference`	**Field Type** string **Description** Required. The name of the parameter. When `type` is `InputTriggeringRecordInitial` or `InputTriggeringRecordUpdated`, the value for `leftValueReference`

Field Name	Description
	must be `$Record`. When `type` is `ScheduledPath`, the value for `leftValueReference` must be `ScheduledPathApiName`.
`type`	**Field Type** FlowTestParameterType (enumeration of type string) **Description** Required. The type of parameter. Possible values are: • `InputTriggeringRecordInitial` • `InputTriggeringRecordUpdated` • `ScheduledPath`—Available in API version 56.0 and later.
`value`	**Field Type** FlowTestReferenceOrValue **Description** Required. The value that the operator applies to the resource reference in the `leftValueReference` field.

Declarative Metadata Sample Definition

The following is an example of a FlowTest component.

```
<?xml version="1.0" encoding="UTF-8"?>
<FlowTest xmlns="http://soap.sforce.com/2006/04/metadata">
    <flowApiName>Example_Test</flowApiName>
    <label>Test Two</label>
    <testPoints>
        <elementApiName>Start</elementApiName>
        <parameters>
            <leftValueReference>$Record</leftValueReference>
            <type>InputTriggeringRecordInitial</type>
            <value>

<sobjectValue>{&amp;quot;AnnualRevenue&amp;quot;:100000,&amp;quot;BillingCity&amp;quot;:&amp;quot;New
York&amp;quot;}}</sobjectValue>
            </value>
        </parameters>
        <parameters>
            <leftValueReference>ScheduledPathApiName</leftValueReference>
            <type>ScheduledPath</type>
            <value>Every_Monday</value>
        </parameters>
        <parameters>
```

```
            <leftValueReference>$Record</leftValueReference>
            <type>InputTriggeringRecordUpdated</type>
            <value>

<sobjectValue>{&quot;AnnualRevenue&quot;:100000,&quot;BillingCity&quot;:&quot;New
York&quot;}</sobjectValue>
            </value>
        </parameters>
    </testPoints>
    <testPoints>
        <assertions>
            <conditions>
                <leftValueReference>$Record.Industry</leftValueReference>
                <operator>EqualTo</operator>
                <rightValue>
                    <stringValue>Other</stringValue>
                </rightValue>
            </conditions>
            <errorMessage>Industry was not set.</errorMessage>
        </assertions>
        <elementApiName>Finish</elementApiName>
    </testPoints>
</FlowTest>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
<types>
<members>Test_Two</members>
<name>FlowTest</name>
</types>
<version>55.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

Folder

Represents a folder. This type extends the Metadata metadata type and inherits its `fullName` field.

Five folder types currently exist in Salesforce:

- Document folder
- Email folder (available for Salesforce Classic email templates only)
- Email Template folder
- Report folder
- Dashboard folder

Folder type names end with the "Folder" suffix. For example, the type name of a document folder is "DocumentFolder".

File Suffix and Directory Location

Folders are stored in the corresponding component directory of the package. These directories are named `documents`, `email`, `emailTemplates`, `reports`, and `dashboards`. Folders don't have a text file representation—they're containers for files. For each folder, an accompanying metadata file named *`FolderName.folderType`*`-meta.xml` is created at the same directory level. The *`FolderName.folderType`*`-meta.xml` metadata file contains the metadata information for that folder, such as the `accessType`. For example, for a documents folder named sampleFolder, there's a *`sampleFolder.documentFolder-meta.xml`* within the `documents` folder of the package.

Version

Folders are available in API version 11.0 and later.

Fields

This metadata type contains the following fields:

Field Name	Field Type	Description
`accessType`	FolderAccessTypes (enumeration of type string)	Required. The type of access for this folder. Valid values are: • `Shared`. This folder is accessible only by the specified set of users. • `Public`. This folder is accessible by all users, including portal users. • `PublicInternal`. This folder is accessible by all users, excluding portal users. This setting is available for report and dashboard folders in organizations with a partner portal or Customer Portal enabled. • `Hidden`. This folder is hidden from all users.
`fullName`	string	The name used as a unique identifier for API access. The `fullName` can contain only underscores and alphanumeric characters. It must be unique, begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. This field is inherited from the Metadata component.
`name`	string	Required. The name of the document folder.
`publicFolderAccess`	PublicFolderAccess (enumeration of type string)	If `Public` is the value for `accessType`, this field indicates the type of access all users have to the contents of the folder. Valid values include: • `ReadOnly`. All users can read the contents of the folder, but no user can change the contents. • `ReadWrite`. All users can read or change the contents of the folder.
`sharedTo`	SharedTo	Sharing access for the folder. See Sharing Considerations in Salesforce Help.

Declarative Metadata Sample Definition

The following is the package manifest definition of a document folder that contains a document:

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <fullName>basic</fullName>
    <types>
        <members>sampleFolder</members>
        <members>sampleFolder/TestDocument.txt</members>
        <name>Document</name>
    </types>
    <version>64.0</version>
</Package>
```

The following is an example of the `sampleFolder-meta.xml` metadata file for the sampleFolder document folder:

```
<?xml version="1.0" encoding="UTF-8"?>
<DocumentFolder xmlns="http://soap.sforce.com/2006/04/metadata">
    <accessType>Public</accessType>
    <name>sampleFolder</name>
    <publicFolderAccess>ReadWrite</publicFolderAccess>
</DocumentFolder>
```

Wildcard Support in the Manifest File

This metadata type doesn't support the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

1. FolderShare
   Represents the settings for enhanced analytics folder sharing. Users can control access to reports or dashboards by giving others Viewer, Editor, or Manager access to the folder that contains the report or dashboard.

SEE ALSO:

- Dashboard
- Document
- EmailTemplate
- Report

FolderShare

Represents the settings for enhanced analytics folder sharing. Users can control access to reports or dashboards by giving others Viewer, Editor, or Manager access to the folder that contains the report or dashboard.

Important: During package installation, FolderShare for DashboardFolder and ReportFolder is ignored.

File Suffix and Directory Location

FolderShare objects are stored in the `reports` and `dashboards` directories. For each report or dashboard folder it contains, there's a metadata file named *`FolderName`*`-meta.xml`. The *`FolderName`*`-meta.xml` metadata file contains the metadata information for that folder, such as the `accessLevel`. For example, if the `reports` directory contains a reports folder named `myReportsFolder`, it also has a *`myReportsFolder-meta.xml`* file at the same level as `myReportsFolder`.

Version

FolderShare components are available in API version 28 and later.

Fields

Field Name	Field Type	Description
`accessLevel`	FolderShareAccessLevel (enumeration of type string)	Required. Specifies the combination of actions that can be taken on the folder. Valid values are: • `View`. User can run a report or refresh a dashboard, but can't edit them. All users have at least Viewer access to report and dashboard folders that have been shared with them. (Some users can have administrative permissions that give them greater access.) • `EditAllContents`. Users can view and modify the reports or dashboards in the folder, and move them to and from any other folders that they have equivalent access to. • `Manage`. Users can do everything Viewers and Editors can do, plus control other users' access to a folder.
`sharedTo`	string	Required. Specifies the user, group, or role that has the specified access level to the folder.
`sharedToType`	FolderSharedToType (enumeration of type string)	Required. Specifies the type of entity that the folder is shared with. Valid values are: • `Group`. Users in a specified public group have the specified access level to the folder. • `Manager`. Available in API version 29.0 and later. • `ManagerAndSubordinatesInternal`. Available in API version 29.0 and later. • `Role`. Users with a specified role have the specified access level to the folder. • `RoleAndSubordinates`. Users with a specified role, and users with a role subordinate to that role, have the specified access level to the folder. The availability of this value depends on the type of org, org creation date, release update enforcement status, and whether digital experiences is enabled. For more information, see this knowledge article. • `RoleAndSubordinatesInternal`. Users with a specified role and users with a role subordinate to that role, except public

Field Name	Field Type	Description
		portal users, have the specified access level to the folder. The availability of this value depends on the type of org, org creation date, release update enforcement status, and whether digital experiences is enabled. For more information, see this knowledge article. • `Organization`. All internal users have the specified access level to the folder. • `Territory`. Users in a specified territory have the specified access level to the folder. • `TerritoryAndSubordinates`. Users in a specified territory, and users in territories subordinate to the specified territory, have the specified access level to the folder. • `AllPrmUsers`. All PRM Portal users have the specified level of access to the folder. • `User`. The specified individual user has the specified level of access to the folder. • `PartnerUser`. The specified individual user of a partner portal has the specified level of access to the folder. • `AllCspUsers`. All Customer Success Portal users have the specified level of access to the folder. • `CustomerPortalUser`. The specified individual user of a customer portal has the specified level of access to the folder. • `PortalRole`. Users with a specified role in a portal have the specified access level to the folder. • `PortalRoleAndSubordinates`. Portal users with a specified role, and portal users with a role subordinate to that role, have the specified access level to the folder.

Declarative Metadata Sample Definition

The following is an example of a FolderShare component for a dashboard folder:

```
<?xml version="1.0" encoding="UTF-8"?>
<DashboardFolder xmlns="http://soap.sforce.com/2006/04/metadata">
  <folderShares>
        <accessLevel>View</accessLevel>
        <sharedTo>R1</sharedTo>
        <sharedToType>Role</sharedToType>
    </folderShares>
</DashboardFolder>
```

Here's an example of a FolderShare component for a report folder:

```
<?xml version="1.0" encoding="UTF-8"?>
<ReportFolder xmlns="http://soap.sforce.com/2006/04/metadata">
  <folderShares>
```

```
        <accessLevel>View</accessLevel>
         <sharedTo>R1</sharedTo>
         <sharedToType>Role</sharedToType>
    </folderShares>
</ReportFolder>
```

Wildcard Support in the Manifest File

This metadata type doesn't support the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ForecastingFilter

Represents the custom filter for including or excluding data from opportunity forecasts.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

ForecastingFilter components have the suffix `.forecastingFilter` and are stored in the `forecastingFilters` folder.

Version

ForecastingFilter components are available in API version 55.0 and later.

Special Access Rules

There are no additional access requirements that are specific to this type.

Fields

Field Name	Description
`filterLogic`	**Field Type** string **Description** The logic that controls the evaluation of conditions. Only `AND` is supported. For example, `1 AND 2 AND 3`.
`forecastingType`	**Field Type** string

Field Name	Description
	Description Required. The ID of the forecast type. Can be linked only to forecast types created in Summer '21 and later.
`forecastingTypeSource`	**Field Type** string **Description** Required. The ID of the forecast type source. Can be linked only to forecast type sources created in Summer '21 or later and with a forecast source definition with source object of 'Opportunity.'
`masterLabel`	**Field Type** string **Description** Required. The label for this object, which displays in Setup. The label is in the default language locale for the organization. If there's no default language locale, the label is in en_US.

Declarative Metadata Sample Definition

The following is an example of a ForecastingFilter component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ForecastingFilter xmlns="http://soap.sforce.com/2006/04/metadata">
    <filterLogic>1 AND 2</filterLogic>
    <forecastingType>d</forecastingType>
    <forecastingTypeSource>d7</forecastingTypeSource>
    <masterLabel>FF_OpportunityLineItem</masterLabel>
</ForecastingFilter>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>ForecastingFilter</name>
    </types>
    <version>55.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ForecastingFilterCondition

Represents the custom filter condition logic for including or excluding data from opportunity forecasts.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

ForecastingFilterCondition components have the suffix `.ForecastingFilterCondition` and are stored in the `ForecastingFilterConditions` folder.

Version

ForecastingFilterCondition components are available in API version 55.0 and later.

Special Access Rules

There are no additional access requirements that are specific to this type.

Fields

Field Name	Description
`fieldName`	**Field Type** string **Description** Required. The name of the opportunity field to be filtered.
`forecastingFilter`	**Field Type** string **Description** Required. The ID of the forecast filter.
`forecastingSourceDefinition`	**Field Type** string **Description** The ID of the forecasting source definition.
`masterLabel`	**Field Type** string

Field Name	Description
	Description Required. The label for this object, which displays in Setup. The label is in the default language locale for the organization. If there's no default language locale, the label is in en_US.
`operation`	**Field Type** FilterOperation (enumeration of type string) **Description** Required. The operator in the filter condition. Possible values are: • `equals` • `greaterOrEqual`—greater than or equal to • `greaterThan` • `lessOrEqual`—less than or equal to • `lessThan` • `notEqual`—not equal to
`sortOrder`	**Field Type** int **Description** Required. The index value for the condition. This value represents the condition in the `FilterLogic` field on the ForecastingFilter object. For example, `1`.
`value`	**Field Type** string **Description** The value of the filter condition. If multiple values are specified, they must be separated by a comma delimiter. Note: If you have multiple currencies enabled, and add a custom filter on a currency field as part of your forecast type definition, the corporate currency at the time the filter was created is used. If you have a single currency enabled, the absolute value is used in your filter condition.

Declarative Metadata Sample Definition

The following is an example of a ForecastingFilterCondition component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ForecastingFilterCondition xmlns="http://soap.sforce.com/2006/04/metadata">
    <colName>mostlikely</colName>
    <fieldName>Amount</fieldName>
    <forecastingFilter>d</forecastingFilter>
    <forecastingSourceDefinition>d7</forecastingSourceDefinition>
    <masterLabel>FFC_Opportunity</masterLabel>
    <operation>greaterThan</masterLabel>
```

```
    <sortOrder>1</masterLabel>
    <value>100000</value>
</ForecastingFilterCondition>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>ForecastingFilterCondition</name>
    </types>
    <version>55.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

Usage

A forecast type can contain up to three filter conditions.

ForecastingSourceDefinition

Represents the object, measure, date type, and hierarchy that a forecast uses to project sales.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

`ForecastingSourceDefinition` components have the suffix `.forecastingSourceDefinition` and are stored in the `forecastingSourceDefinitions` folder.

Version

ForecastingSourceDefinition components are available in API version 52.0 and later.

Fields

Field Name	Field Type	Description
categoryField	string	Name of the forecast category that is associated with the forecast type. Possible values are: • Opportunity.ForecastCategoryName
dateField	string	Field that is used for the forecast type's date type. For example, the CloseDate field on Opportunity is used for opportunity close date-based forecast types. Possible values are: • Opportunity.CloseDate • OpportunityLineItem.ServiceDate • OpportunityLineItemSchedule.ScheduleDate
familyField	string	Use this field to group forecasts by product family. Possible values are: • Product2.Family
masterLabel	string	Required. Controlling label for this forecasting source definition.
measureField	string	Field that is used for the forecast type's measure. For example, the Amount field on Opportunity is associated with revenue-based forecast types. Possible values are*: • Opportunity.Amount • Opportunity.***Custom*** • Opportunity.TotalOpportunityQuantity • OpportunityLineItem.***Custom*** • OpportunityLineItem.Quantity • OpportunityLineItem.TotalPrice • OpportunityLineItemSchedule.***Custom*** • OpportunityLineItemSchedule.Quantity • OpportunityLineItemSchedule.Revenue • OpportunitySplit.***Custom*** • OpportunitySplit.SplitAmount *Where *Custom* represents the name of the custom field that a forecast type's measure is based on. Example: Use Megawatts__c to forecast energy consumption.
sourceObject	string	Required. Object associated with this forecasting source definition. Possible values are: • Opportunity • OpportunityLineItem • OpportunityLineItemSchedule • OpportunitySplit

Field Name	Field Type	Description
		• `Product2`
`territory2Field`	string	For a territory-based forecast type, indicates the field that is used for territory information. Possible values are: • `Opportunity.Territory2Id` For user role-based forecast types, this value is `null`.
`userField`	string	Specifies who owns the forecast. Possible values are: • `Opportunity.OwnerId` • `OpportunitySplit.SplitOwnerId`

Declarative Metadata Sample Definition

The following is an example of a ForecastingSourceDefinition component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ForecastingSourceDefinition xmlns="http://soap.sforce.com/2006/04/metadata">
    <masterLabel>TestFsd</masterLabel>
    <sourceObject>Opportunity</sourceObject>
    <measureField>Opportunity.Amount</measureField>
    <dateField>Opportunity.CloseDate</dateField>
    <userField>Opportunity.OwnerId</userField>
    <categoryField>Opportunity.ForecastCategoryName</categoryField>
</ForecastingSourceDefinition>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>ForecastingSourceDefinition</name>
    </types>
    <version>52.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

Usage

- Forecast types that were available before API version 52.0 can be activated, deactivated, and deleted but not created. To enable an existing forecast type, update the active flag.

- Forecast types that are available only in API version 52.0 and later can be created, activated, deactivated, and deleted. If the forecast type doesn't exist, it is created in the inactive state. If the forecast type exists, the active flag is updated. Deploy the zip file twice to create and activate the forecast type.
- Deploy Metadata API types in the following sequence: ForecastingSettings, ForecastingType, ForecastingSourceDefinition, and then ForecastingTypeSource. If all are specified in the package file, the sequence is followed automatically.

ForecastingType

Represents a forecast type.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

`ForecastingType` components have the suffix `.forecastingType` and are stored in the `forecastingTypes` folder.

Version

ForecastingType components are available in API version 52.0 and later.

Fields

Field Name	Field Type	Description
`active`	boolean	Required. If `true`, the forecast type is active. If `false`, the forecast type isn't active. The default value is `false`.
`amount`	boolean	Required. If `true`, the forecast type is based on a revenue measure. If `false`, the forecast type is based on a quantity measure. The default value is `true`.
`dateType`	string	Required. The date type that forecast amounts are based on. • `OpportunityCloseDate`: Base forecasts on opportunity close dates. • `ProductDate`: Base forecasts on opportunity product line item dates, if available. • `ScheduleDate`: Base forecasts on opportunity product schedule dates, if available. The following values are available in API version 52.0 and later, in Performance Edition and in Unlimited Edition with Sales Cloud.

Field Name	Field Type	Description
		• `OLIMeasureCloseDateOnly`: Base forecasts on opportunity close dates. • `ProductDateOnly`: Base forecasts on opportunity product line item dates, if available. • `ScheduleDateOnly`: Base forecasts on opportunity product schedule dates, if available.
`developerName`	string	Required. The name of the forecasting type. The `DeveloperName` is called `name` in ForecastingSettings on page 1954 and Forecasting Type in custom reports.
`forecastingGroupDeveloperName`	string	Indicates the forecast group assigned to the forecast type. Required if `hasCustomGroup` is `true`.
`hasCustomGroup`	boolean	Indicates whether the forecasting type has a forecast group, based on a custom picklist assigned. Use ForecastingGroup and ForecastingGroupItems subtypes in ForecastingSettings to identify the group and the values.
`hasProductFamily`	boolean	Required. If `true`, the forecast type includes product families. If `false`, the forecast type doesn't include product families. The default value is `false`.
`masterLabel`	string	Required. Controlling label for this ForecastingType value. This display value is the internal label that doesn't get translated.
`opportunitySplitType`	string	Indicates whether the forecasting type has a split type and, if so, the name of the split type.
`opptyLineItemSplitType`	string	Indicates whether the forecasting type has an opportunity line item (product) split type and, if so, the name of the line item split type. Available in API version 58.0 and later.
`quantity`	boolean	Required. If `true`, the forecast type is based on a quantity measure. If `false`, the forecast type is based on a revenue measure. The default value is `false`.
`roleType`	string	Required. Indicates whether the role type has a ForecastingType, and if so, which ForecastingType. Possible values are `R` (user role-based forecast type) and `Y` (Territory2-based forecast type).
`territory2Model`	string	Indicates whether the ForecastingType has a Territory2 model and, if so, the name of the Territory2 model.

Declarative Metadata Sample Definition

The following is an example of a ForecastingType component using the role hierarchy.

```
<?xml version="1.0" encoding="UTF-8"?>
<ForecastingType xmlns="http://soap.sforce.com/2006/04/metadata">
```

```
    <active>false</active>
    <amount>true</amount>
    <dateType>0</dateType>
    <developerName>qqw</developerName>
    <hasProductFamily>false</hasProductFamily>
    <masterLabel>qqw</masterLabel>
    <quantity>false</quantity>
    <roleType>R</roleType>
</ForecastingType>
```

The following is an example of a ForecastingType component using the territory hierarchy.

```
<?xml version="1.0" encoding="UTF-8"?>
<ForecastingType xmlns="http://soap.sforce.com/2006/04/metadata">
    <active>false</active>
    <amount>false</amount>
    <dateType>0</dateType>
    <developerName>New_Model6</developerName>
    <hasProductFamily>false</hasProductFamily>
    <masterLabel>Opportunity Quantity by Territory</masterLabel>
    <quantity>true</quantity>
    <roleType>Y</roleType>
    <territory2Model>New_Model6</territory2Model>
</ForecastingType>
```

The following is an example of a ForecastingType component using an opportunity split type.

```
<?xml version="1.0" encoding="UTF-8"?>
<ForecastingType xmlns="http://soap.sforce.com/2006/04/metadata">
    <active>false</active>
    <amount>true</amount>
    <dateType>0</dateType>
    <developerName>split12</developerName>
    <hasProductFamily>false</hasProductFamily>
    <masterLabel>split12</masterLabel>
    <opportunitySplitType>Custom_Revenue</opportunitySplitType>
    <quantity>false</quantity>
    <roleType>R</roleType>
</ForecastingType>
```

The following is an example of a ForecastingType component using an opportunity line item split type.

```
<?xml version="1.0" encoding="UTF-8"?>
<ForecastingType xmlns="http://soap.sforce.com/2006/04/metadata">
    <active>true</active>
    <amount>true</amount>
    <dateType>0</dateType>
    <developerName>productrevenuesplit</developerName>
    <hasProductFamily>true</hasProductFamily>
    <masterLabel>productrevenuesplit</masterLabel>
    <opportunitySplitType>Revenue</opportunitySplitType>
    <opptyLineItemSplitType>Revenue</opptyLineItemSplitType>
    <quantity>false</quantity>
    <roleType>R</roleType>
</ForecastingType>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>ForecastingType</name>
    </types>
    <version>52.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

Usage

- Legacy forecast types that were available before API version 52.0 can be deactivated but not activated, created, or deleted.
- Forecast types that are available only in API version 52.0 and later can be created, activated, deactivated, and deleted. If the forecast type doesn't exist, it's created in the inactive state. If the forecast type exists, the active flag is updated. Deploy the zip file twice to create and activate the forecast type.
- Deploy Metadata API types in this sequence: ForecastingSettings, ForecastingType, ForecastingSourceDefinition, and then ForecastingTypeSource. If all are specified in the package file, the sequence is followed automatically.

ForecastingTypeSource

Represents the mapping of a forecasting source definition to a forecast type.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

`ForecastingTypeSource` components have the suffix `.forecastingTypeSource` and are stored in the `ForecastingTypeSources` folder.

Version

ForecastingTypeSource components are available in API version 52.0 and later.

Fields

Field Name	Field Type	Description
forecastingSourceDefinition	string	Required. ID of the forecasting source definition.
forecastingType	string	Required. ID of the forecast type. Can be linked only to forecast types created in Summer '21 and later.
masterLabel	string	Required. Controlling label for this forecasting type source.
parentSourceDefinition	string	For forecast types not based on the Opportunity object and not based on a custom measure, this value represents the parent ForecastingSourceDefinition of the linked ForecastingSourceDefinition. • Opportunity Product is the parent of Opportunity. • Opportunity Split is the parent of Opportunity. • Line Item Schedule is the parent of Opportunity Product.
relationField	string	Represents the field that links the source objects of the parent ForecastingSourceDefinition to the child ForecastingSourceDefinition. Possible values are: • OpportunityLineItem.OpportunityId • OpportunityLineItem.Product2Id • OpportunityLineItemSchedule.OpportunityLineItemId • OpportunitySplit.OpportunityId
sourceGroup	int	Required. Represents a grouping of forecasting source definitions.

Declarative Metadata Sample Definition

The following are two examples of a ForecastingTypeSource component. The first bases forecasts on the Opportunity Product object. The second bases forecasts on the Line Item Schedule object.

```
<?xml version="1.0" encoding="UTF-8"?>
<ForecastingTypeSource xmlns="http://soap.sforce.com/2006/04/metadata">
    <forecastingSourceDefinition>FSD_OpportunityLineItem</forecastingSourceDefinition>
    <forecastingType>d</forecastingType>
    <masterLabel>ForecastingTypeSource_d7</masterLabel>
    <parentSourceDefinition>FSD_OpportunityLineItemSchedule1</parentSourceDefinition>
    <relationField>OpportunityLineItemSchedule.OpportunityLineItemId</relationField>
    <sourceGroup>1</sourceGroup>
</ForecastingTypeSource>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<ForecastingTypeSource xmlns="http://soap.sforce.com/2006/04/metadata">
<forecastingSourceDefinition>FSDOpportunityLineItemSchedule</forecastingSourceDefinition>

    <forecastingType>c3</forecastingType>
    <masterLabel>ForecastingTypeSource_c37syR</masterLabel>
```

```
    <sourceGroup>1</sourceGroup>
</ForecastingTypeSource>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>ForecastingTypeSource</name>
    </types>
    <version>52.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

Usage

- Forecast types that were available before API version 52.0 can be activated, deactivated, and deleted but not created. To enable an existing forecast type, update the active flag.
- Forecast types that are available only in API version 52.0 and later can be created, activated, deactivated, and deleted. If the forecast type doesn't exist, it is created in the inactive state. If the forecast type exists, the active flag is updated. Deploy the zip file twice to create and activate the forecast type.
- Deploy Metadata API types in the following sequence: ForecastingSettings, ForecastingType, ForecastingSourceDefinition, and then ForecastingTypeSource. If all are specified in the package file, the sequence is followed automatically.

FuelType

Represents a custom fuel type in an org.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

FuelType components have the suffix `.fuelType` and are stored in the `fuelTypes` folder.

Version

FuelType components are available in API version 57.0 and later.

Special Access Rules

The Net Zero Cloud permission set license is required to access this object along with the user access for carbon accounting and org access for custom fuels and unit of measures (UOMs).

Fields

Field Name	Description
`description`	**Field Type** string **Description** Description about the fuel type.
`isActive`	**Field Type** boolean **Description** Indicates whether the fuel type is active (`true`) or not (`false`). The default value is `false`.
`isProtected`	**Field Type** boolean **Description** An auto-generated value that doesn't impact the behavior of the metadata type. The default value is `false`.
`isStationaryAssetFuel`	**Field Type** boolean **Description** Indicates whether the fuel type is used in stationary assets (`true`) or not (`false`). The default value is `false`.
`isVehicleAssetFuel`	**Field Type** boolean **Description** Indicates whether the fuel type is used in a vehicle asset (`true`) or not (`false`). The default value is `false`.
`masterLabel`	**Field Type** string **Description** Required.

Field Name	Description
	A user-friendly name for FuelType, which is defined when the FuelType is created.

Declarative Metadata Sample Definition

The following is an example of a FuelType component.

```
<?xml version="1.0" encoding="UTF-8"?>
<FuelType xmlns="http://soap.sforce.com/2006/04/metadata">
    <description>This is Petrol Fuel Type</description>
    <isProtected>true</isProtected>
    <isActive>true</isActive>
    <isStationaryAssetFuel>true</isStationaryAssetFuel>
    <isVehicleAssetFuel>true</isVehicleAssetFuel>
    <masterLabel>Petrol</masterLabel>
</FuelType>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
   <types>
      <members>Petrol</members>
      <members>Diesel</members>
      <members>Kerosine</members>
      <name>FuelType</name>
   </types>
   <version>57.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

FuelTypeSustnUom

Represents a mapping between the custom fuel types and their corresponding unit of measure (UOM) values defined by a customer in an org.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

FuelTypeSustnUom components have the suffix `.fuelTypeSustnUom` and are stored in the `fuelTypeSustnUoms` folder.

Version

FuelTypeSustnUom components are available in API version 57.0 and later.

Special Access Rules

The Net Zero Cloud permission set license is required to access this object along with the user access for carbon accounting and org access for custom fuels and UOMs.

Fields

Field Name	Description
`fuelType`	**Field Type** string **Description** Required. The name of the fuel type that's mapped to the unit of measure. Possible values are: • `AutogasLPG` • `Biodiesel` • `Biomass` • `CityGas` • `CompressedNaturalGasCNG` • `Cooling` • `Diesel` • `Electricity` • `Ethanol` • `FuelOil` • `Gasoline` • `Heat` • `HeavyOil` • `ITElectricity` • `JetFuel` • `Kerosene` • `LightOil` • `LiquidNaturalGasLNG`

Field Name	Description
	• `MobileDiesel` • `NaturalGas` • `Propane` • `Refrigerant` • `Steam`
`isProtected`	**Field Type** boolean **Description** An auto-generated value that doesn't impact the behavior of the metadata type. The default value is `false`.
`masterLabel`	**Field Type** string **Description** A user-friendly name for FuelTypeSustnUom, which is defined when the FuelTypeSustnUom is created.
`unitOfMeasure`	**Field Type** string **Description** Required. The unit of measure that's mapped to the fuel type. Possible values are: • `1000m3` • `GJ` • `GWh` • `Kiloliters` • `Liters` • `MJ` • `MMBtu` • `MWh` • `Therms` • `Tonnes` • `UkGallons` • `UsGallons` • `ccf` • `kG` • `kWh`

Field Name	Description
	• kcal • lbs • longTons • m3 • shortTons

Declarative Metadata Sample Definition

The following is an example of a FuelTypeSustnUom component.

```
<?xml version="1.0" encoding="UTF-8"?>
<FuelTypeSustnUom xmlns="http://soap.sforce.com/2006/04/metadata">
    <fuelType>FuelOil</fuelType>
    <isProtected>false</isProtected>
    <masterLabel>FuelOil_Liters</masterLabel>
    <unitOfMeasure>Liters</unitOfMeasure>
</FuelTypeSustnUom>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <fullName>Pkg</fullName>
    <types>
        <members>FuelOil_Liters</members>
        <members>Gas_1000m3</members>
        <members>Heat_kWh</members>
        <name>FuelTypeSustnUom</name>
    </types>
    <version>57.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

FunctionReference

Represents information about a deployed Salesforce Function that can be invoked from the org. This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

FunctionReference does not support direct access and should be managed using Salesforce CLI commands associated with Functions. A FunctionReference component file has the suffix `.functions` and is stored in the `functions` directory.

Version

FunctionReference components are available in API version 52.0 and later.

Special Access Rules

FunctionReference components can't be used directly. Always use Salesforce CLI commands associated with Functions to properly deploy Functions and associate Functions with orgs. Attempting to manipulate FunctionReference components directly without using Functions CLI commands is not supported.

Fields

Field Name	Field Type	Description
`description`	string	Represents the description of the Salesforce Function.
`label`	string	Represents the label for the Salesforce Function.
`permissionSet`	string	Represents a set of permissions that's used to control org resources that the Function has access to.

FundraisingConfig

Represents a collection of settings to configure the fundraising product.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

FundraisingConfig components have the suffix `.fundraisingConfig` and are stored in the `fundraisingConfigs` folder.

Version

FundraisingConfig components are available in API version 58.0 and later.

Special Access Rules

Your org must have Fundraising Access license as a part of the Nonprofit Cloud to access this object.

Fields

Field Name	Description
`donorMatchingMethod`	**Field Type** DonorMatchingMethod (enumeration of type string) **Description** Reserved for future use.
`failedTransactionCount`	**Field Type** int **Description** The count of consecutive failed past transactions before the gift commitment status is changed to Failing. If set to 0, the status is never auto-changed to Failing.
`householdSoftCreditRole`	**Field Type** string **Description** Reserved for future use.
`installmentExtDayCount`	**Field Type** int **Description** The duration in the number of days before or after an unpaid transaction in a gift commitment is marked as another installment in the gift commitment schedule. The unpaid transaction within the grace period is considered a gift transaction.
`isHshldSoftCrAutoCrea`	**Field Type** boolean **Description** Reserved for future use.
`lapsedUnpaidTrxnCount`	**Field Type** int **Description** The count of consecutive unpaid past transactions before the gift commitment status is changed to Lapsed. If set to 0, the status is never auto-changed to Lapsed.
`masterLabel`	**Field Type** string **Description** A user-friendly name for FundraisingConfig, which is defined when the FundraisingConfig is created.

Field Name	Description
`shouldClosePaidRcrCmt`	**Field Type** boolean **Description** Indicates whether to automatically close a recurring gift commitment when it has no ongoing or future schedule and no unpaid transaction (`true`) or not (`false`). The default value is `false`. Available in API version 59.0 and later.
`shouldCreateRcrSchdTrxn`	**Field Type** boolean **Description** Indicates whether the next transaction in a recurring schedule is automatically created (`true`) or not (`false`). The default value is `true`. Available in API version 59.0 and later.
`utmCampaignSrcObj`	**Field Type** string **Description** Name of the sObject of the campaign for which the donation was received. Available in API version 64.0 and later.
`utmCampaignSrcObjField`	**Field Type** string **Description** Name of the field on the sObject in `utmCampaignSrcObj` of the campaign for which the donation was received. Available in API version 64.0 and later.
`utmMediumSrcObj`	**Field Type** string **Description** Name of the sObject that stores data about the message channel from which the donation originated. Available in API version 64.0 and later.
`utmMediumSrcObjField`	**Field Type** string **Description** Name of the field on the sObject in `utmMediumSrcObj` that stores data about the message channel from which the donation originated. Available in API version 64.0 and later.
`utmSourceSrcObj`	**Field Type** string

Field Name	Description
	Description Name of the sObject that stores data about the source of a donation. Available in API version 64.0 and later.
`utmSourceSrcObjField`	**Field Type** string **Description** Name of the field on the sObject in `utmSourceSrcObj` that stores data about the source of a donation. Available in API version 64.0 and later.

Declarative Metadata Sample Definition

The following is an example of a FundraisingConfig component.

```
<?xml version="1.0" encoding="UTF-8"?>
<FundraisingConfig xmlns="http://soap.sforce.com/2006/04/metadata">
    <lapsedUnpaidTrxnCount>5</lapsedUnpaidTrxnCount>
    <householdSoftCreditRole>Admin</householdSoftCreditRole>
    <isHshldSoftCrAutoCrea>true</isHshldSoftCrAutoCrea>
    <installmentExtDayCount>7</installmentExtDayCount>
    <donorMatchingMethod>No_Matching</donorMatchingMethod>
    <donorExternalIdField>TestExtId__c</donorExternalIdField>
    <failedTransactionCount>12</failedTransactionCount>
    <outreachSourceCodeGenFmla>OutreachSourceCodeGenFmla</outreachSourceCodeGenFmla>
    <shouldCreateRcrSchdTrxn>true</shouldCreateRcrSchdTrxn>
    <shouldClosePaidRcrCmt>false</shouldClosePaidRcrCmt>
    <masterLabel>MasterLabel</masterLabel>
    <utmMediumSrcObj>UTM Medium Src Obj</utmMediumSrcObj>
    <utmMediumSrcObjField>UTM Medium Src Obj Field</utmMediumSrcObjField>
    <utmSourceSrcObj>UTM Source Src Obj</utmSourceSrcObj>
    <utmSourceSrcObjField>UTM Source Src Obj Field</utmSourceSrcObjField>
    <utmCampaignSrcObj>UTM Campaign Src Obj</utmCampaignSrcObj>
    <utmCampaignSrcObjField>UTM Campaign Src Obj Field</utmCampaignSrcObjField>
</FundraisingConfig>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>FundraisingConfig</name>
    </types>
    <version>64.0</version>
</Package>
```

GatewayProviderPaymentMethodType

Represents an entity that allows integrators and payment providers to choose an active payment to receive an order's payment data rather than allowing the Salesforce Order Management platform to select a default payment method. This object is available in API version 51 and later.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Version

gatewayProviderPaymentMethodType components are available in API version 51 and later.

Special Access Rules

Fields

Field Name	Description
`comment`	**Field Type** textarea **Description** Additional details about the gateway provider payment method type record. Max length is 1000 characters.
`gtwyProvPaymentMethodType`	**Field Type** string **Description** Links the Salesforce payment method to the payment method used in the Salesforce Order Management storefront. Your payment gateway integration uses this field when finding a payment method to link to a payment. The value of `GtwyProviderPaymentMethodType` must match the payment method value sent to the order's Payment Instrument in Salesforce Order Management. Here are examples of payment method values that Salesforce could receive from Salesforce Order Management. • `CREDIT_CARD` • `BASIC_CREDIT` • `CreditCard` • `GooglePay` • `ApplePay`
`masterLabel`	**Field Type** string

Field Name	Description
	Description Required. The gateway provider payment method type name that appears in the user interface.
paymentGatewayProvider	**Field Type** reference **Description** Specifies the payment gateway provider that Salesforce Order Management should use when processing payments. One payment gateway provider can be related to multiple payment method types.
paymentMethodType	**Field Type** picklist **Description** Specifies the type of payment method used on an order in Salesforce Order Management. Possible values are: • AlternativePaymentMethod • CardPaymentMethod • DigitalWallet
recordType	**Field Type** reference **Description** ID of the record type entity related to the gateway provider payment method type. This is a relationship field.

Declarative Metadata Sample Definition

The following is an example of a GatewayProviderPaymentMethodType component.

```
<?xml version="1.0" encoding="UTF-8"?>
<GatewayProviderPaymentMethodType xmlns="http://soap.sforce.com/2006/04/metadata">
    <gtwyProviderPaymentMethodType>Klarna</gtwyProviderPaymentMethodType>
    <masterLabel>Test</masterLabel>
    <paymentGatewayProvider>adyen__Adyen</paymentGatewayProvider>
    <paymentMethodType>AlternativePaymentMethod</paymentMethodType>
    <recordType>AlternativePaymentMethod.Klarna</recordType>
</GatewayProviderPaymentMethodType>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
```

```
    <types>
        <members>*</members>
        <name>GatewayProviderPaymentMethodType</name>
    </types>
    <version>51.0</version>
</Package>
```

GenAiFunction

Represents an agent action that can be added to an AI agent.

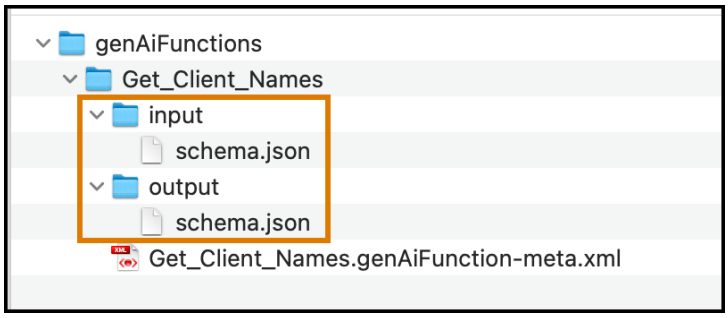
Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

GenAiFunction components have the suffix `.genAiFunction` and are stored in the `genAiFunctions` folder.

GenAiFunction components can contain folders for the input on page 1247 and output on page 1249 schemas. Here's an example component, showing the schema folders.



See the Input Folder on page 1247 and Output Folder on page 1249 sections for more information.

Version

GenAiFunction components are available in API version 60.0 and later.

Special Access Rules

GenAiFunction is available only if Agents is enabled in your org.

Fields

Field Name	Description
`description`	**Field Type** string **Description** A description explaining the general purpose and domain of the action.
`invocationTarget`	**Field Type** string **Description** Required. Target invocation used by invocation operations.
`invocationTargetType`	**Field Type** PlannerFunctionInvocableTargetType (enumeration of type string) **Description** Required. Invocable action type used by invocation operations. Values are: • `apex` • `api` • `createCatalogItemRequest` • `flow` • `generatePromptResponse` • `externalService` • `quickAction` • `slack` • `standardInvocableAction`
`isConfirmationRequired`	**Field Type** boolean **Description** Indicates whether confirmation is required for this action.
`isIncludeInProgressIndicator`	**Field Type** boolean **Description** Indicates whether to display the progress indicator for this action.
`mappingAttributes`	**Field Type** GenAiPlannerAttr[]

Field Name	Description
	Description List of attributes for the planner.
`masterLabel`	**Field Type** string **Description** Required. The master label for the generative AI action.
`progressIndicatorMessage`	**Field Type** string **Description** The progress message.

GenAiPlannerAttr

Field Name	Description
`description`	**Field Type** string **Description** Description of the planner attribute.
`label`	**Field Type** string **Description** Required. Label for the planner attribute.
`name`	**Field Type** string **Description** Required. Name of the planner attribute.
`parameterName`	**Field Type** string **Description** Required. The parameter name.
`parameterType`	**Field Type** PlannerAttrMappingType (enumeration of type string)

Field Name	Description
	Description Required. The parameter type. Values are: • `input` • `output`

Input Folder

The `input` folder contains a `schema.json` file with the action inputs. Here's a sample input schema file.

```
{
  "required" : ["OwnerId", "Status"],
  "properties" : {
    "OwnerId" : {
      "title" : "Owner Id",
      "description" : "ID of the Salesforce record that owns the request.",
      "lightning:type" : "lightning__textType",
      "lightning:isPII" : false,
      "copilotAction:isUserInput" : true
    },
    "Status" : {
      "title" : "Request Status",
      "description" : "The status of the contact request.",
      "lightning:type" : "lightning__textType",
      "lightning:isPII" : false,
      "copilotAction:isUserInput" : true
    }
  },
  "lightning:type" : "lightning__objectType"
}
```

This table describes the properties that you can specify in this JSON file.

Field Name	Description
`required`	**Field Type** array of strings **Description** A list of all the required properties in the input list.
`properties`	**Field Type** object **Description** The parent object for all the properties associated with the input.
`{PROPERTY_NAME}` (child of properties field)	**Field Type** object

Field Name	Description
	Description Required. Each property in the `properties` object must contain the API value corresponding to the input parameter.
`title` (child of properties.PROPERTY_NAME field)	**Field Type** string **Description** Required. The label for the property.
`description` (child of properties.PROPERTY_NAME field)	**Field Type** string **Description** The description for the property.
`lightning:type` (child of properties.PROPERTY_NAME field)	**Field Type** string **Description** Required. Describes the type of property. Some of the possible values include: • `lightning__booleanType`—Boolean type. • `lightning__dateType`—Date type. • `lightning__dateTimeType`—Date and time type. When using this value, you must provide a `dateTime` string. This field is specified in the format `yyyy-MM-dd'T'HH:mm:ss.SSSZ`. You can also provide a `timeZone` string. Use the timezone attribute to specify a time zone in the IANA time zone database format. • `lightning__integerType`—Integer type. When using this value, you can also provide `maximum` and `minimum` numbers for this property. • `lightning__listType`—List type. • `lightning__multilineTextType`—Multi-line text type. When using this value, you can also provide `maxLength` and `minLength` numbers for this property. • `lightning__numberType`—Number type. When using this value, you can also provide `maximum` and `minimum` numbers for this property. • `lightning__objectType`—Object type. When using this value, you must also provide a `properties` object that contains the list of subproperties within it.

Field Name	Description
	• `lightning__recordIdType`—Record ID type. • `lightning__richTextType`—Rich text type. When using this value, you can also provide `maxLength` and `minLength` numbers for this property. • `lightning__textType`—Text type. When using this value, you can also provide `maxLength` and `minLength` numbers for this property. The maximum text type length is 250 characters. • `lightning__urlType`—URL type. When using this value, you can also provide an array of strings in a `lightning:allowedUrlSchemes` field to specify the list of allowed URL schemes.
`lightning:isPII` (child of properties.PROPERTY_NAME field)	**Field Type** boolean **Description** Indicates whether the property contains personally identifiable information.
`copilotAction:isUserInput` (child of properties.PROPERTY_NAME field)	**Field Type** boolean **Description** Indicates whether the property is presented as user input.
`lightning:type`	**Field Type** string **Description** Required. Describes the type of input. This value should always be `lightning__objectType`.

Output Folder

The `output` folder contains a `schema.json` file with the action output. Here's a sample output schema file.

```
{
  "properties" : {
    "Id" : {
      "title" : "Contact Request Id",
      "description" : "ID of the Salesforce contact request record.",
      "lightning:type" : "lightning__recordIdType",
      "lightning:isPII" : false,
      "copilotAction:isDisplayable" : true,
      "copilotAction:isUsedByPlanner" : true
    }
  },
```

```
  "lightning:type" : "lightning__objectType"
}
```

This table describes the properties that you can specify in this JSON file.

Field Name	Description
`properties`	**Field Type** object **Description** The parent object for all the properties associated with the output.
`{PROPERTY_NAME}` (child of properties field)	**Field Type** object **Description** Required. Each property in the `properties` object must contain the API value corresponding to the output parameter.
`title` (child of properties.PROPERTY_NAME field)	**Field Type** string **Description** Required. The label for the property.
`description` (child of properties.PROPERTY_NAME field)	**Field Type** string **Description** The description for the property.
`lightning:type` (child of properties.PROPERTY_NAME field)	**Field Type** string **Description** Required. Describes the type of output. Some of the possible values include: • `lightning__booleanType`—Boolean type. • `lightning__dateType`—Date type. • `lightning__dateTimeType`—Date and time type. When using this value, you must provide a `dateTime` string. This field is specified in the format `yyyy-MM-dd'T'HH:mm:ss.SSSZ`. You can also provide a `timeZone` string. Use the timezone attribute to specify a time zone in the IANA time zone database format.

Field Name	Description
	• `lightning__integerType`—Integer type. When using this value, you can also provide `maximum` and `minimum` numbers for this property. • `lightning__listType`—List type. • `lightning__multilineTextType`—Multi-line text type. When using this value, you can also provide `maxLength` and `minLength` numbers for this property. • `lightning__numberType`—Number type. When using this value, you can also provide `maximum` and `minimum` numbers for this property. • `lightning__objectType`—Object type. When using this value, you must also provide a `properties` object that contains the list of subproperties within it. • `lightning__recordIdType`—Record ID type. • `lightning__richTextType`—Rich text type. When using this value, you can also provide `maxLength` and `minLength` numbers for this property. • `lightning__textType`—Text type. When using this value, you can also provide `maxLength` and `minLength` numbers for this property. The maximum text type length is 250 characters. • `lightning__urlType`—URL type. When using this value, you can also provide an array of strings in a `lightning:allowedUrlSchemes` field to specify the list of allowed URL schemes.
`lightning:isPII` (child of properties.PROPERTY_NAME field)	**Field Type** boolean **Description** Indicates whether the property contains personally identifiable information.
`copilotAction:isDisplayable` (child of properties.PROPERTY_NAME field)	**Field Type** boolean **Description** Indicates whether the property is displayable as output.
`copilotAction:isUsedByPlanner` (child of properties.PROPERTY_NAME field)	**Field Type** boolean

Field Name	Description
	Description Indicates whether the property is used by the agent planner. At least one output property must have this value as true or else the planner returns random responses.
`lightning:type`	**Field Type** string **Description** Required. Describes the type of output. This value should always be `lightning__objectType`.

Declarative Metadata Sample Definition

The following is an example of a GenAiFunction component.

```
<?xml version="1.0" encoding="UTF-8"?>
<GenAiFunction xmlns="http://soap.sforce.com/2006/04/metadata">
    <description>get tracking information</description>
    <invocationTarget>TrackShipment</invocationTarget>
    <invocationTargetType>apex</invocationTargetType>
    <isConfirmationRequired>false</isConfirmationRequired>
    <masterLabel>get_tracking_info</masterLabel>
</GenAiFunction>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
   <types>
      <members>*</members>
      <name>GenAiFunction</name>
   </types>
   <version>60.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

GenAiPlanner

Represents a planner for an agent. It's a container for all the topics and actions used to interact with a large language model (LLM).

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

GenAiPlanner components have the suffix `.genAiPlanner` and are stored in the `genAiPlanners` folder.

Version

GenAiPlanner components are available in API version 60.0 to 63.0. GenAiPlannerBundle replaces GenAiPlanner in API version 64.0 and later.

Special Access Rules

GenAiPlanner is available only if Agents is enabled in your org.

Fields

Field Name	Description
`attributeMappings`	**Field Type** GenAiPlannerAttrMapping[] **Description** A list of action attribute mappings.
`capabilities`	**Field Type** string **Description** A set of tags associated with the agent.
`description`	**Field Type** string **Description** A description explaining the general purpose and domain of the agent.
`genAiFunctions`	**Field Type** GenAiPlannerFunctionDef[] **Description** A list of agent action definitions, such as a knowledge action, that are not contained in a topic.
`genAiPlannerRuleExpressions`	**Field Type** GenAiPlannerRuleExpr[]

Field Name	Description
	Description Deprecated. Use `ruleExpressions` instead.
`genAiPlugins`	**Field Type** GenAiPlannerFunctionDef[] **Description** A list of agent topic definitions.
`masterLabel`	**Field Type** string **Description** Required. The master label for the planner.
`plannerType`	**Field Type** PlannerType (enumeration of type string) **Description** Required. A particular approach to problem solving that is given as prompt instructions to a large language model (LLM). The only supported value is: • `AiCopilot__ReAct`—Uses a reactive reasoning strategy to solve problems with the LLM. This strategy consists of prompting the LLM to generate the next step in response to an event and the current context. It differs from a sequential reasoning engine in that it doesn't plan more than one step ahead of time.
`ruleExpressionAssignments`	**Field Type** GenAiPlannerRuleExprAsgn[] **Description** A list of rule expression assignments.
`ruleExpressions`	**Field Type** GenAiPlannerRuleExprDef[] **Description** A list of rule expressions.

GenAiPlannerAttrMapping

Represents an attribute mapping, which enables you to map the output of one action attribute to the input of another attribute. This mapping enables you to propagate sensitive data safely without relying on untrusted user input.

Field Name	Description
`attributeName`	**Field Type** string **Description** Required. The attribute name in the format: `Namespace.TopicName.ActionName.AttributeName`.
`attributeType`	**Field Type** AttributeType (enumeration of type string) **Description** Required. The attribute type. Values are: • `CustomPluginFunctionAttribute`—Map to a custom action input or output • `StandardPluginFunctionInput`—Map to a standard action input • `StandardPluginFunctionOutput`—Map output to a variable
`constantValue`	**Field Type** string **Description** Reserved for future use.
`mappingTargetName`	**Field Type** string **Description** The target name for the attribute mapping.
`mappingType`	**Field Type** AttributeMappingType (enumeration of type string) **Description** Required. The target type. Values are: • `ActionAttribute` • `Constant` • `Variable` • `ContextVariable`

GenAiPlannerFunctionDef

Represents an agent topic or action definition.

Field Name	Description
genAiCustomizedPlugin	**Field Type** GenAiLocalPlugin[] **Description** A list of custom agent topics.
genAiFunctionName	**Field Type** string **Description** The name of the agent action.
genAiPluginName	**Field Type** string **Description** The name of the agent topic.

GenAiLocalPlugin

Represents a custom agent topic.

Field Name	Description
aiPluginUtterances	**Field Type** AiPluginUtteranceDef[] **Description** A list of utterances that can be used to pick a topic during runtime.
description	**Field Type** string **Description** The description of the topic.
genAiFunctions	**Field Type** GenAiPluginFunctionDef[] **Description** A list of functions in the topic.
genAiPluginInstructions	**Field Type** GenAiPluginInstructionDef[] **Description** A list of instructions in the topic.

Field Name	Description
`language`	**Field Type** string **Description** Required. The language of the topic.
`masterLabel`	**Field Type** string **Description** Required. The master label for the topic.
`name`	**Field Type** string **Description** Required. Represents the API name of the topic. This name must be unique across all custom and customized topics. Can contain only underscores and alphanumeric characters and must be unique in your org. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
`pluginType`	**Field Type** PluginType (enumeration of type string) **Description** Required. Values are: • `Topic`
`scope`	**Field Type** string **Description** A specific job description for a topic.

GenAiPlannerRuleExprAsgn

Represents a rule-expression assignment to either a topic or an action.

Field Name	Description
`ruleExpressionName`	**Field Type** string **Description** Required. The name of the rule expression.
`targetName`	**Field Type** string **Description** Required. The target of the assignment, which is a `Namespace.TopicName.ActionName` or a `TopicName`.
`targetType`	**Field Type** string **Description** Required. The type of the target. Values are: • `Function`—A knowledge action • `Plugin`—A topic • `PluginFunction`—An action in a topic

GenAiPlannerRuleExprDef

Represents a rule expression, which conditionally locks or unlocks topics and actions based on defined security criteria.

Field Name	Description
`conditions`	**Field Type** GenAiPlannerRuleExprCondition[] **Description** A list of conditions for a rule expression.
`expression`	**Field Type** string **Description** An expression with the combined conditions.
`expressionLabel`	**Field Type** string **Description** Required. The expression label.

Field Name	Description
`expressionName`	**Field Type** string **Description** Required. The expression name.
`expressionType`	**Field Type** string **Description** The expression type. Values are: • `handlebars`—Reserved for future use • `sel`—Salesforce Expression Language, as used in formula fields

GenAiPlannerRuleExprCondition

Represents a condition for a rule expression.

Field Name	Description
`leftOperand`	**Field Type** string **Description** Required. The left operand in the expression.
`leftOperandType`	**Field Type** GenAiAgentVariableType (enumeration of type string) **Description** Required. The type for the left operand. Values are: • `Variable` • `ContextVariable` • `Attribute`
`operator`	**Field Type** GenAiRuleExpressionOperator (enumeration of type string) **Description** Required. The operator in the expression. Values are: • `equal` • `greaterThan` • `greaterThanOrEqual` • `lessThan` • `lessThanOrEqual`

Field Name	Description
	• notEqual • isEmpty • isNotEmpty
rightOperandValue	**Field Type** string **Description** The value for the right operand.

Declarative Metadata Sample Definition

The following is an example of a GenAiPlanner component.

```
<?xml version="1.0" encoding="UTF-8"?>
<GenAiPlanner xmlns="http://soap.sforce.com/2006/04/metadata">
    <description>Copilot reasoning engine description</description>
    <masterLabel>EmployeeCopilotPlanner</masterLabel>
    <plannerType>AiCopilot__SequentialPlannerIntentClassifier</plannerType>
</GenAiPlanner>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
   <types>
      <members>*</members>
      <name>GenAiPlanner</name>
   </types>
   <version>60.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

GenAiPlannerBundle

Represents a planner for an agent or agent template. It's a container for all the topics and actions used to interact with a large language model (LLM).

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

GenAiPlannerBundle components have the suffix `.genAiPlannerBundle` and are stored in a subfolder for the agent in the `genAiPlannerBundles` folder.

Version

GenAiPlannerBundle components are available in API version 64.0 and later. GenAiPlannerBundle replaces GenAiPlanner, which is available in API version 63.0 and earlier.

Special Access Rules

GenAiPlannerBundle is available only if Agents is enabled in your org.

Fields

Field Name	Description
`attributeMappings`	**Field Type** GenAiPlannerAttrMapping[] **Description** A list of action attribute mappings.
`botTemplate`	**Field Type** string **Description** If this planner is used by an agent template instead of an agent, this field is the template associated with the planner.
`capabilities`	**Field Type** string **Description** A set of tags associated with the agent.
`description`	**Field Type** string **Description** A description explaining the general purpose and domain of the agent.
`genAiFunctions`	**Field Type** GenAiPlannerFunctionDef[] **Description** A list of agent action definitions, such as a knowledge action, that are not contained in a topic.

Field Name	Description
genAiPlugins	**Field Type** GenAiPlannerFunctionDef[] **Description** A list of agent topic definitions.
masterLabel	**Field Type** string **Description** Required. The master label for the planner.
plannerType	**Field Type** PlannerType (enumeration of type string) **Description** Required. A particular approach to problem solving that is given as prompt instructions to a large language model (LLM). The supported values are: • AiCopilot__AgileAppDev—Uses an iterative development strategy to assist with building applications using the LLM. This strategy prompts the LLM to generate modular, testable components based on evolving user input and context. Unlike linear workflows, it supports continuous refinement and feedback loops throughout the development process. • AiCopilot__ReAct—Uses a reactive reasoning strategy to solve problems with the LLM. This strategy consists of prompting the LLM to generate the next step in response to an event and the current context. It differs from a sequential reasoning engine in that it doesn't plan more than one step ahead of time.
ruleExpressionAssignments	**Field Type** GenAiPlannerRuleExprAsgn[] **Description** A list of rule expression assignments.
ruleExpressions	**Field Type** GenAiPlannerRuleExprDef[] **Description** A list of rule expressions.

GenAiPlannerAttrMapping

Represents an attribute mapping, which enables you to map the output of one action attribute to the input of another attribute. This mapping enables you to propagate sensitive data safely without relying on untrusted user input.

Field Name	Description
`attributeName`	**Field Type** string **Description** Required. The attribute name in the format: `Namespace.TopicName.ActionName.AttributeName`.
`attributeType`	**Field Type** AttributeType (enumeration of type string) **Description** Required. The attribute type. Values are: • `CustomPluginFunctionAttribute`—Map to a custom action input or output • `StandardPluginFunctionInput`—Map to a standard action input • `StandardPluginFunctionOutput`—Map output to a variable
`constantValue`	**Field Type** string **Description** Reserved for future use.
`mappingTargetName`	**Field Type** string **Description** The target name for the attribute mapping.
`mappingType`	**Field Type** AttributeMappingType (enumeration of type string) **Description** Required. The target type. Values are: • `ActionAttribute` • `Constant` • `Variable` • `ContextVariable`

GenAiPlannerFunctionDef

Represents an agent topic or action definition.

Field Name	Description
genAiCustomizedPlugin	**Field Type** GenAiLocalPlugin[] **Description** A list of custom agent topics.
genAiFunctionName	**Field Type** string **Description** The name of the agent action.
genAiPluginName	**Field Type** string **Description** The name of the agent topic.

GenAiLocalPlugin

Represents a custom agent topic.

Field Name	Description
aiPluginUtterances	**Field Type** AiPluginUtteranceDef[] **Description** A list of utterances that can be used to pick a topic during runtime.
canEscalate	**Field Type** boolean **Description** Determines whether this topic is applicable for escalation to a rep.
description	**Field Type** string **Description** Required. The description of the topic.
genAiFunctions	**Field Type** GenAiPluginFunctionDef[]

Field Name	Description
	Description A list of functions in the topic.
`genAiPluginInstructions`	**Field Type** GenAiPluginInstructionDef[] **Description** A list of instructions in the topic.
`language`	**Field Type** string **Description** Required. The language of the topic.
`masterLabel`	**Field Type** string **Description** Required. The master label for the topic.
`name`	**Field Type** string **Description** Required. Represents the API name of the topic. This name must be unique across all custom and customized topics. Can contain only underscores and alphanumeric characters and must be unique in your org. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
`pluginType`	**Field Type** PluginType (enumeration of type string) **Description** Required. Values are: • `Topic` • `APICustomTopic`
`scope`	**Field Type** string

Field Name	Description
	Description A specific job description for a topic.

GenAiPlannerRuleExprAsgn

Represents a rule-expression assignment to either a topic or an action.

Field Name	Description
`ruleExpressionName`	**Field Type** string **Description** Required. The name of the rule expression.
`targetName`	**Field Type** string **Description** Required. The target of the assignment, which is a `Namespace.TopicName.ActionName` or a `TopicName`.
`targetType`	**Field Type** string **Description** Required. The type of the target. Values are: • `Function`—A knowledge action • `Plugin`—A topic • `PluginFunction`—An action in a topic

GenAiPlannerRuleExprDef

Represents a rule expression, which conditionally locks or unlocks topics and actions based on defined security criteria.

Field Name	Description
`conditions`	**Field Type** GenAiPlannerRuleExprCondition[] **Description** A list of conditions for a rule expression.

Field Name	Description
`expression`	**Field Type** string **Description** An expression with the combined conditions.
`expressionLabel`	**Field Type** string **Description** Required. The expression label.
`expressionName`	**Field Type** string **Description** Required. The expression name.
`expressionType`	**Field Type** string **Description** The expression type. Values are: • `handlebars`—Reserved for future use • `sel`—Salesforce Expression Language, as used in formula fields

GenAiPlannerRuleExprCondition

Represents a condition for a rule expression.

Field Name	Description
`leftOperand`	**Field Type** string **Description** Required. The left operand in the expression.
`leftOperandType`	**Field Type** GenAiAgentVariableType (enumeration of type string) **Description** Required. The type for the left operand. Values are: • `Variable` • `ContextVariable` • `Attribute`

Field Name	Description
operator	**Field Type** GenAiRuleExpressionOperator (enumeration of type string) **Description** Required. The operator in the expression. Values are: • equal • greaterThan • greaterThanOrEqual • lessThan • lessThanOrEqual • notEqual • isEmpty • isNotEmpty
rightOperandValue	**Field Type** string **Description** The value for the right operand.

Declarative Metadata Sample Definition

Here's an example of a GenAiPlannerBundle component.

```
<?xml version="1.0" encoding="UTF-8"?>
<GenAiPlannerBundle xmlns="http://soap.sforce.com/2006/04/metadata">
    <attributeMappings>

attributeName>SvcCopilotTmpl__CaseManagement.SvcCopilotTmpl__CreateCaseEnhancedData.verifiedCustomerID</attributeName>

        <attributeType>StandardPluginFunctionInput</attributeType>
        <mappingTargetName>VerifiedCustomerId</mappingTargetName>
        <mappingType>Variable</mappingType>
    </attributeMappings>
    <attributeMappings>

<attributeName>SvcCopilotTmpl__CaseManagement.SvcCopilotTmpl__GetCasesForVerifiedContact.verifiedContactID</attributeName>

        <attributeType>StandardPluginFunctionInput</attributeType>
        <mappingTargetName>VerifiedCustomerId</mappingTargetName>
        <mappingType>Variable</mappingType>
    </attributeMappings>
    <attributeMappings>

<attributeName>SvcCopilotTmpl__CaseManagement.SvcCopilotTmpl__GetCaseByVerifiedCaseNumber.verifiedContactID</attributeName>

        <attributeType>StandardPluginFunctionInput</attributeType>
```

```
        <mappingTargetName>VerifiedCustomerId</mappingTargetName>
        <mappingType>Variable</mappingType>
    </attributeMappings>
    <attributeMappings>

<attributeName>SvcCopilotTmpl__ServiceCustomerVerification.SvcCopilotTmpl__SendEmailVerificationCode.authenticationKey</attributeName>

        <attributeType>StandardPluginFunctionOutput</attributeType>
        <mappingTargetName>authenticationKey</mappingTargetName>
        <mappingType>Variable</mappingType>
    </attributeMappings>
    <attributeMappings>

<attributeName>SvcCopilotTmpl__ServiceCustomerVerification.SvcCopilotTmpl__SendEmailVerificationCode.customerId</attributeName>

        <attributeType>StandardPluginFunctionOutput</attributeType>
        <mappingTargetName>customerId</mappingTargetName>
        <mappingType>Variable</mappingType>
    </attributeMappings>
    <attributeMappings>

<attributeName>SvcCopilotTmpl__ServiceCustomerVerification.SvcCopilotTmpl__SendEmailVerificationCode.customerType</attributeName>

        <attributeType>StandardPluginFunctionOutput</attributeType>
        <mappingTargetName>customerType</mappingTargetName>
        <mappingType>Variable</mappingType>
    </attributeMappings>
    <attributeMappings>

<attributeName>SvcCopilotTmpl__ServiceCustomerVerification.SvcCopilotTmpl__VerifyCustomer.authenticationKey</attributeName>

        <attributeType>StandardPluginFunctionInput</attributeType>
        <mappingTargetName>authenticationKey</mappingTargetName>
        <mappingType>Variable</mappingType>
    </attributeMappings>
    <attributeMappings>

<attributeName>SvcCopilotTmpl__ServiceCustomerVerification.SvcCopilotTmpl__VerifyCustomer.customerId</attributeName>

        <attributeType>StandardPluginFunctionInput</attributeType>
        <mappingTargetName>customerId</mappingTargetName>
        <mappingType>Variable</mappingType>
    </attributeMappings>
    <attributeMappings>

<attributeName>SvcCopilotTmpl__ServiceCustomerVerification.SvcCopilotTmpl__VerifyCustomer.customerType</attributeName>

        <attributeType>StandardPluginFunctionInput</attributeType>
        <mappingTargetName>customerType</mappingTargetName>
        <mappingType>Variable</mappingType>
    </attributeMappings>
    <attributeMappings>

<attributeName>SvcCopilotTmpl__ServiceCustomerVerification.SvcCopilotTmpl__VerifyCustomer.isVerified</attributeName>
```

```
        <attributeType>StandardPluginFunctionOutput</attributeType>
        <mappingTargetName>isVerified</mappingTargetName>
        <mappingType>Variable</mappingType>
    </attributeMappings>
    <attributeMappings>

<attributeName>SvcCopilotTmpl__ServiceCustomerVerification.SvcCopilotTmpl__VerifyCustomer.customerId</attributeName>

        <attributeType>StandardPluginFunctionOutput</attributeType>
        <mappingTargetName>VerifiedCustomerId</mappingTargetName>
        <mappingType>Variable</mappingType>
    </attributeMappings>
    <attributeMappings>

<attributeName>SvcCopilotTmpl__AccountManagement.SvcCopilotTmpl__ResetSecurePassword.verifiedContactID</attributeName>

        <attributeType>StandardPluginFunctionInput</attributeType>
        <mappingTargetName>VerifiedCustomerId</mappingTargetName>
        <mappingType>Variable</mappingType>
    </attributeMappings>
    <attributeMappings>

<attributeName>SvcCopilotTmpl__AccountManagement.SvcCopilotTmpl__UpdateVerifiedContact.verifiedContactID</attributeName>

        <attributeType>StandardPluginFunctionInput</attributeType>
        <mappingTargetName>VerifiedCustomerId</mappingTargetName>
        <mappingType>Variable</mappingType>
    </attributeMappings>
    <description>Deliver personalized customer interactions with an autonomous AI agent.
Agentforce Service Agent intelligently supports your customers with common inquiries and
escalates complex issues.</description>
    <genAiFunctions>
       <genAiFunctionName>EmployeeCopilot__AnswerQuestionsWithKnowledge</genAiFunctionName>

    </genAiFunctions>
    <genAiPlugins>
        <genAiPluginName>SvcCopilotTmpl__AccountManagement</genAiPluginName>
    </genAiPlugins>
    <genAiPlugins>
        <genAiPluginName>SvcCopilotTmpl__CaseManagement</genAiPluginName>
    </genAiPlugins>
    <genAiPlugins>
        <genAiPluginName>SvcCopilotTmpl__Escalation</genAiPluginName>
    </genAiPlugins>
    <genAiPlugins>
        <genAiPluginName>SvcCopilotTmpl__GeneralFAQ</genAiPluginName>
    </genAiPlugins>
    <genAiPlugins>
        <genAiPluginName>SvcCopilotTmpl__ServiceCustomerVerification</genAiPluginName>
    </genAiPlugins>
    <masterLabel>ASA Template Base</masterLabel>
    <plannerType>AiCopilot__ReAct</plannerType>
    <ruleExpressionAssignments>
```

```
        <ruleExpressionName>Verified_User</ruleExpressionName>
        <targetName>SvcCopilotTmpl__AccountManagement</targetName>
        <targetType>Plugin</targetType>
    </ruleExpressionAssignments>
    <ruleExpressionAssignments>
        <ruleExpressionName>Verified_User</ruleExpressionName>
        <targetName>SvcCopilotTmpl__CaseManagement</targetName>
        <targetType>Plugin</targetType>
    </ruleExpressionAssignments>
    <ruleExpressions>
        <conditions>
            <leftOperand>isVerified</leftOperand>
            <leftOperandType>Variable</leftOperandType>
            <operator>equal</operator>
            <rightOperandValue>true</rightOperandValue>
        </conditions>
        <expression>Verified_User</expression>
        <expressionLabel>Verified User</expressionLabel>
        <expressionName>Verified_User</expressionName>
        <expressionType>sel</expressionType>
    </ruleExpressions>
</GenAiPlannerBundle>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
   <types>
      <members>*</members>
      <name>GenAiPlannerBundle</name>
   </types>
   <version>64.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

GenAiPlugin

Represents an agent topic, which is a category of actions related to a particular job to be done by AI agents.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

GenAiPlugin components have the suffix `.genAiPlugin` and are stored in the `genAiPlugins` folder.

Version

GenAiPlugin components are available in API version 62.0 and later.

Special Access Rules

GenAiPlugin is available only if Agents is enabled in your org.

Fields

Field Name	Description
`aiPluginUtterances`	**Field Type** AiPluginUtteranceDef[] **Description** A list of utterances that can be used to pick a topic during runtime.
`canEscalate`	**Field Type** boolean **Description** Indicates whether this topic is eligible for escalation to a rep.
`description`	**Field Type** string **Description** The description of the topic.
`developerName`	**Field Type** string **Description** Required. Represents the API name of the topic. This name must be unique across all custom and customized topics. Can contain only underscores and alphanumeric characters and must be unique in your org. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
`genAiFunctions`	**Field Type** GenAiPluginFunctionDef[] **Description** A list of functions in the topic.
`genAiPluginInstructions`	**Field Type** GenAiPluginInstructionDef[]

Field Name	Description
	Description A list of instructions in the topic.
`language`	**Field Type** string **Description** Required. The language of the topic.
`masterLabel`	**Field Type** string **Description** Required. The master label for the topic.
`pluginType`	**Field Type** PluginType (enumeration of type string) **Description** Required. Values are: • `Topic` • `APICustomTopic`
`scope`	**Field Type** string **Description** A specific job description for a topic.

GenAiPluginFunctionDef

A function in the topic.

Field Name	Description
`functionName`	**Field Type** string **Description** Required. The API name of the function.

Declarative Metadata Sample Definition

The following is an example of a GenAiPlugin component.

```
<GenAiPlugin xmlns="http://soap.sforce.com/2006/04/metadata"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <description>Engages and interacts with the user about any request that could be CRM
data related.
     This could be tasks such as identify and summarize records, answer queries, aggregate
 data,
     find and query objects, update records, or drafting and refining emails.</description>

    <developerName>General_CRM_Customized</developerName>
    <genAiFunctions>
        <functionName>EmployeeCopilot__IdentifyObjectByName</functionName>
    </genAiFunctions>
    <genAiFunctions>
        <functionName>EmployeeCopilot__IdentifyRecordByName</functionName>
    </genAiFunctions>
    <genAiFunctions>
        <functionName>EmployeeCopilot__QueryRecords</functionName>
    </genAiFunctions>
    <genAiFunctions>
        <functionName>EmployeeCopilot__QueryRecordsWithAggregate</functionName>
    </genAiFunctions>
    <genAiFunctions>
        <functionName>EmployeeCopilot__GetActivitiesTimeline</functionName>
    </genAiFunctions>
    <genAiFunctions>
        <functionName>EmployeeCopilot__GetActivityDetails</functionName>
    </genAiFunctions>
    <genAiFunctions>
        <functionName>EmployeeCopilot__GetRecordDetails</functionName>
    </genAiFunctions>
    <genAiFunctions>
        <functionName>EmployeeCopilot__DraftOrReviseEmail</functionName>
    </genAiFunctions>
    <genAiFunctions>
        <functionName>EmployeeCopilot__UpdateRecordFields</functionName>
    </genAiFunctions>
    <genAiFunctions>
        <functionName>EmployeeCopilot__WebSearch</functionName>
    </genAiFunctions>
    <genAiFunctions>
        <functionName>EmployeeCopilot__AnswerQuestionsWithKnowledge</functionName>
    </genAiFunctions>
    <genAiPluginInstructions>
       <description>There are multiple available data retrieval functions at your disposal.

          You can use each one of them multiple times if needed. You should use functions

          as many times as necessary until you have all the data required to fulfill the
request of the user.
          You can perform extra calls if you think you can get additional relevant
information.</description>
```

```
        <developerName>therearemu0</developerName>
        <language xsi:nil="true"/>
        <masterLabel>therearemu</masterLabel>
    </genAiPluginInstructions>
    <genAiPluginInstructions>
       <description>Do not declare your intent i.e. &amp;quot;I will now retrieve the data&amp;quot;
 - Just fetch the data.</description>
        <developerName>donotdecla1</developerName>
        <language xsi:nil="true"/>
        <masterLabel>donotdecla</masterLabel>
    </genAiPluginInstructions>
    <genAiPluginInstructions>
        <description>Identify the object type (i.e., leads, opportunities, accounts) the
user asks about.
           If unclear, confirm with the user and make a suggestion based on the query
context and history.</description>
        <developerName>identifyth2</developerName>
        <language xsi:nil="true"/>
        <masterLabel>identifyth</masterLabel>
    </genAiPluginInstructions>
    <genAiPluginInstructions>
        <description>When only the name of a record is mentioned in the user request, you
 MUST call the IdentifyRecordByName action
          to get the necessary IDs.</description>
        <developerName>whenonlyth3</developerName>
        <language xsi:nil="true"/>
        <masterLabel>whenonlyth</masterLabel>
    </genAiPluginInstructions>
    <genAiPluginInstructions>
       <description>Always call QueryRecords &amp;amp; QueryRecordsWithAggregate passing plain
 natural language english as input.
          You must Include the record ID in the input if available.</description>
        <developerName>alwayscall4</developerName>
        <language xsi:nil="true"/>
        <masterLabel>alwayscall</masterLabel>
    </genAiPluginInstructions>
    <genAiPluginInstructions>
        <description>For accounts and contacts, combine the WebSearch action with the CRM
 data retrieval actions.
           MUST maintain citations in the answer.</description>
        <developerName>foraccount5</developerName>
        <language xsi:nil="true"/>
        <masterLabel>foraccount</masterLabel>
    </genAiPluginInstructions>
    <genAiPluginInstructions>
        <description>When a user asks for a summary or overview of a record, use
GetRecordDetails to get an overview
          of the data of the record, then use other data retrieval actions as
needed.</description>
        <developerName>whenausera6</developerName>
        <language xsi:nil="true"/>
        <masterLabel>whenausera</masterLabel>
    </genAiPluginInstructions>
    <genAiPluginInstructions>
```

```
        <description>When asked for a summary on multiple records, you must iterate over
all record IDs and for each one,
          call GetRecordDetails.</description>
        <developerName>whenaskedf7</developerName>
        <language xsi:nil="true"/>
        <masterLabel>whenaskedf</masterLabel>
    </genAiPluginInstructions>
    <genAiPluginInstructions>
        <description>If the user asks about activities you must call
GetActivitiesTimeline.</description>
        <developerName>iftheusera8</developerName>
        <language xsi:nil="true"/>
        <masterLabel>iftheusera</masterLabel>
    </genAiPluginInstructions>
    <genAiPluginInstructions>
        <description>When providing the Activity Types for GetActivitiesTimeline, choose
all the types that are relevant
           to the user request. Examples - User Request 1 - &amp;quot;What questions does John
 Doe have that need addressing?&amp;quot;,
           Activity Types - &amp;quot;Call&amp;quot;, &amp;quot;Email&amp;quot;.
          User Request 2 - &amp;quot;What are the next activities for John Doe?&amp;quot;, Activity
 Types - &amp;quot;Task&amp;quot;, &amp;quot;Event&amp;quot;.</description>
        <developerName>whenprovid9</developerName>
        <language xsi:nil="true"/>
        <masterLabel>whenprovid</masterLabel>
    </genAiPluginInstructions>
    <genAiPluginInstructions>
        <description>When asked about recent activities, you should provide the answer,
starting from the last 30 days
          and ending on the current date, unless a specific date range is specified by the
 user.</description>
        <developerName>whenaskeda10</developerName>
        <language xsi:nil="true"/>
        <masterLabel>whenaskeda</masterLabel>
    </genAiPluginInstructions>
    <genAiPluginInstructions>
        <description>When you are asked about a single Call, Meeting, Email or any other
single specific activity, call GetActivityDetails.</description>
        <developerName>whenyouare11</developerName>
        <language xsi:nil="true"/>
        <masterLabel>whenyouare</masterLabel>
    </genAiPluginInstructions>
    <genAiPluginInstructions>
        <description>Always use DraftOrReviseEmail when asked to generate a new email or
revise a previously generated email.</description>
        <developerName>alwaysused12</developerName>
        <language xsi:nil="true"/>
        <masterLabel>alwaysused</masterLabel>
    </genAiPluginInstructions>
    <genAiPluginInstructions>
        <description>ExtractRecordFieldsAndValuesFromUserInput must be called prior to
UpdateRecordFields.</description>
        <developerName>extractrec13</developerName>
        <language xsi:nil="true"/>
```

```
        <masterLabel>extractrec</masterLabel>
    </genAiPluginInstructions>
    <genAiPluginInstructions>
        <description>Avoid using structured lists, bullet points, or numbered lists.
Instead, present the information in complete sentences
          and paragraphs as if you were writing an article or a report.</description>
        <developerName>avoidusing14</developerName>
        <language xsi:nil="true"/>
        <masterLabel>avoidusing</masterLabel>
    </genAiPluginInstructions>
    <genAiPluginInstructions>
       <description>User questions which can be answered by knowledge articles or documents.
 These questions usually want information,
          instructions or guidance, including but not limited to customer questions about
 company information, policies and frequently asked questions.</description>
        <developerName>userquesti15</developerName>
        <language xsi:nil="true"/>
        <masterLabel>userquesti</masterLabel>
    </genAiPluginInstructions>
    <language>en_US</language>
    <masterLabel>General CRM Customized</masterLabel>
    <pluginType>Topic</pluginType>
    <scope>Your job is to interact and answer questions for the user about anything
Salesforce or CRM data related,
      combining all data retrieval functions. i.e: QueryRecords(), GetRecordDetails(),
GetActivitiesTimeline(), GetActivityDetails(), WebSearch()</scope>
</GenAiPlugin>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
   <types>
      <members>*</members>
      <name>GenAiPlugin</name>
   </types>
   <version>62.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

GenAiPluginInstructionDef

Represents a topic instruction.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

GenAiPluginInstructionDef components are part of a GenAiPlugin component and aren't used separately.

Version

GenAiPluginInstructionDef components are available in API version 62.0 and later.

Special Access Rules

GenAiPluginInstructionDef is available only if Agents is enabled in your org.

Fields

Field Name	Description
`description`	**Field Type** string **Description** Required. Description of the topic instruction.
`developerName`	**Field Type** string **Description** Required. Represents the API name of the topic instruction. Can contain only underscores and alphanumeric characters and must be unique in your org. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
`language`	**Field Type** string **Description** Required. The language of the topic instruction.
`masterLabel`	**Field Type** string **Description** Required. The master label for the topic instruction.

Declarative Metadata Sample Definition

See GenAiPlugin on page 1271.

GenAiPromptTemplate

Represents the definition of a prompt template, including its related objects and fields.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

`GenAiPromptTemplate` components have the suffix `.genAiPromptTemplate` and are stored in the `genAiPromptTemplates` folder.

Version

GenAiPromptTemplate components are available in API version 60.0 and later.

Special Access Rules

GenAiPromptTemplate is available only if Prompt Builder is enabled in your org and you have the Prompt Template Manager permission.

Fields

Field Name	Description
activeVersion	**Field Type** int **Description** This tag will be deprecated in 63.0 and will not work in 64.0 and later. Use activeVersionIdentifier instead.
`activeVersionIdentifier`	**Field Type** string **Description** Specifies the version identifier of the active prompt template version. This tag will use versionIdentifier as the value for the active version.

Field Name	Description
`description`	**Field Type** string **Description** A description of the prompt template.
`masterLabel`	**Field Type** string **Description** Required. A user-friendly name for GenAiPromptTemplate, which is defined when the GenAiPromptTemplate is created.
`relatedEntity`	**Field Type** string **Description** The Salesforce record type that the prompt template is associated with.
`relatedField`	**Field Type** string **Description** The Salesforce field that the prompt template is associated with.
`templateVersions`	**Field Type** GenAiPromptTemplateVersion on page 1281[] **Description** Required. An array of prompt template versions.
`type`	**Field Type** string **Description** Required. Represents the template type that the prompt template is based on. Valid values are: • `einstein_gpt__fieldCompletion` • `einstein_gpt__salesEmail` • `einstein_gpt__recordSummary` • `einstein_gpt__flex` • `einstein_gpt__caseEmailDraft`
`visibility`	**Field Type** GenAiPromptTemplateVisibilityType (enumeration of type string) **Description** Indicates the scope of visibility for the prompt template. Valid values are:

Field Name	Description
	• `API` • `Global`

GenAiPromptTemplateVersion

Represents a version of a prompt template.

Field Name	Description
`content`	**Field Type** string **Description** Required. Text of the prompt template version.
`description`	**Field Type** string **Description** Description of the prompt template version.
`inputs`	**Field Type** GenAiPromptTemplateInput on page 1282[] **Description** An array of prompt template inputs associated with the prompt template version.
`primaryModel`	**Field Type** string **Description** The model associated with the prompt template version.
`status`	**Field Type** GenAiPromptTemplateStatus (enumeration of type string) **Description** Required. Indicates the status of the prompt template in Prompt Builder. Valid values are: • `Published`—Published version of a prompt template. The active version of the prompt template must be published. Published prompt templates can't be edited in Salesforce, but are currently editable from the Metadata API. We recommend that you don't edit published prompt templates from the API. • `Draft`—Draft version of a prompt template.

Field Name	Description
`templateDataProviders`	**Field Type** GenAiPromptTemplateDataProvider on page 1283[] **Description** An array of prompt template data providers associated with the prompt template version.
`versionNumber`	**Field Type** int **Description** Required. Version number of the prompt template version. Versions are counted sequentially from 1. This tag will be deprecated in 63.0 and will not work in 64.0 and later. Use versionIdentifier instead.
`versionIdentifier`	**Field Type** string **Description** Version identifier of the prompt template version identifier. This is generated automatically once a template is deployed and retrieved from an org. If a unique value is not specified then it will be generated for you. It must be unique for each version for a given template.

GenAiPromptTemplateInput

Represents an input for a prompt template, such as a Salesforce record, field, or Apex primitive data type.

Field Name	Description
`apiName`	**Field Type** string **Description** Required. Name of the prompt template input parameter.
`definition`	**Field Type** string **Description** Required. The URI definition of the input parameter. For example, `SOBJECT://Account` and `SOBJECT://Account/Description`.
`description`	**Field Type** string

Field Name	Description
	Description Description of the prompt template input parameter.
`masterLabel`	**Field Type** string **Description** A user-friendly name for GenAiPromptTemplateInput, which is defined when the GenAiPromptTemplateInput is created.
`referenceName`	**Field Type** string **Description** Required. Name of the prompt template input to use in expressions. For example, `Input:Recipient` and `Input:Sender</referenceName>`.
`required`	**Field Type** boolean **Description** Required. Specifies whether this input parameter is required (`true`) or optional (`false`).

GenAiPromptTemplateDataProvider

Represents a source of data for a prompt template version, such as an invocable action, flow, or Apex method.

Field Name	Description
`definition`	**Field Type** string **Description** Required. The URI definition of the data provider, such as `flow://ns__CallToActionFlow`.
`parameters`	**Field Type** GenAiPromptTemplateDataProviderParam on page 1284[] **Description** An array of parameters associated with the data provider.
`referenceName`	**Field Type** string **Description** Required. Name of the data provider to use in expressions.

GenAiPromptTemplateDataProviderParam

Represents a parameter that a data provider uses to retrieve information.

Field Name	Description
`definition`	**Field Type** string **Description** Required. URI definition of the parameter. For example, `SOBJECT://User</definition>`.
`isRequired`	**Field Type** boolean **Description** Required. Specifies whether the parameter is required (`true`) or optional (`false`).
`parameterName`	**Field Type** string **Description** Required. Name of the parameter.
`valueExpression`	**Field Type** string **Description** Value or expression of the parameter to use in prompt template text. For example, `{!$Input:Recipient}`.

Declarative Metadata Sample Definition

The following is an example of a GenAiPromptTemplate component.

```
<?xml version="1.0" encoding="UTF-8"?>
<GenAiPromptTemplate xmlns="http://soap.sforce.com/2006/04/metadata">

<activeVersionIdentifier>a94ACO8feseESrBtllQUNLvKEqlCHiPSDLX5rXUnvPs=_1</activeVersionIdentifier>

    <description>Recommend relevant financial products to client based on their needs and
 goals</description>
    <masterLabel>Recommend Relevant Products</masterLabel>
    <relatedEntity>Contact</relatedEntity>
    <templateVersions>
        <content>You are a financial advisor at {!$Input:Sender.CompanyName} and your name
 is {!$Input:Sender.Name}. You are writing an email to a prospective client recommending
relevant financial products based on their data and goals. Reference the data below to
generate your email, recommending only relevant products for the customer that match our
recommendation criteria for each product.
Client name: {!$Input:Recipient.Name}
```

```
Client age: {!$Input:Recipient.Age__c}
Client occupation: {!$Input:Recipient.Occupation__c}
Client income: {!$Input:Recipient.Income__c}
Client financial goals: {!$Input:Recipient.Financial_Goals__c}

{!$Flow:Fetch_Products.Prompt}

Generate a subject line that can increase the open rate using words and content that is
related to the email body content. It must be no more than 10 words.
Start the opening message of the email with an ice-breaker talking about relevant challenges
 or opportunities with personal finance and how you can help.
Indirectly allude to a point of common interest, shared background, or relevant experience
 with {!$Input:Recipient.Name}. You aim to subtly reference or highlight this connection
to establish rapport, demonstrate relevance, and foster a sense of familiarity.
Indirectly encourage the lead {!$Input:Recipient.Name} to respond to your email by showing
 that you are willing to discuss opportunities for working together and answer any questions
 they may have.
Be concise in your email.</content>
        <inputs>
            <apiName>Sender</apiName>
            <definition>SOBJECT://User</definition>
            <referenceName>Input:Sender</referenceName>
            <required>true</required>
        </inputs>
        <inputs>
            <apiName>Recipient</apiName>
            <definition>SOBJECT://Contact</definition>
            <referenceName>Input:Recipient</referenceName>
            <required>true</required>
        </inputs>
        <primaryModel>sfdc_ai__DefaultOpenAIGPT4</primaryModel>
        <status>Published</status>
        <templateDataProviders>
            <definition>flow://Fetch_Products</definition>
            <parameters>
                <definition>SOBJECT://User</definition>
                <isRequired>true</isRequired>
                <parameterName>Sender</parameterName>
                <valueExpression>{!$Input:Sender}</valueExpression>
            </parameters>
            <parameters>
                <definition>SOBJECT://Contact</definition>
                <isRequired>true</isRequired>
                <parameterName>Recipient</parameterName>
                <valueExpression>{!$Input:Recipient}</valueExpression>
            </parameters>
            <referenceName>Flow:Fetch_Products</referenceName>
        </templateDataProviders>
        <versionIdentifier>a94ACO8feseESrBtllQUNLvKEqlCHiPSDLX5rXUnvPs=_1
</versionIdentifier>
    </templateVersions>
    <type>einstein_gpt__salesEmail</type>
    <visibility>Global</visibility>
</GenAiPromptTemplate>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>GenAiPromptTemplate</name>
    </types>
    <version>60.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

GenAiPromptTemplateActv

Represents the activation status of a Salesforce-provided prompt template.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

`GenAiPromptTemplateActv` components have the suffix `.genAiPromptTemplateActivation` and are stored in the `genAiPromptTemplateActivations` folder.

Version

GenAiPromptTemplateActv components are available in API version 60.0 and later.

Special Access Rules

GenAiPromptTemplate is available only if Prompt Builder is enabled in your org and you have the Prompt Template Manager permission.

Fields

Field Name	Description
`accessLevel`	**Field Type** GenAiPromptTemplateActvAccessLevel (enumeration of type string)

Field Name	Description
	Description Indicates which users can access the Salesforce-provided prompt template. Valid values are: • `Allowed`—Users with access to Prompt Builder can see the prompt template. • `Blocked`—Only admin users with access to Prompt Builder can see the prompt template.
`developerName`	**Field Type** string **Description** Name of the activation record. This name can contain only underscores and alphanumeric characters. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
`masterLabel`	**Field Type** string **Description** A user-friendly name for GenAiPromptTemplateActv, which is defined when the GenAiPromptTemplateActv is created.
`templateDeveloperName`	**Field Type** string **Description** Name of the Salesforce-provided prompt template that the activation record is associated with.

Declarative Metadata Sample Definition

The following is an example of a GenAiPromptTemplateActv component.

```
<?xml version="1.0" encoding="UTF-8"?>
<GenAiPromptTemplateActv xmlns="http://soap.sforce.com/2006/04/metadata">
    <MasterLabel> Activation Record for Prompt Template </MasterLabel>
    <PromptTemplateDeveloperName>einstein_gpt__introductionLeadEmail
    </PromptTemplateDeveloperName>
    <DeveloperName>HideIntroductionLeadEmail</DeveloperName>
    <Description>Status of template </Description>
    <AccessLevel>BLOCKED</AccessLevel>
</GenAiPromptTemplateActv>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <name>GenAiPromptTemplateActv</name>
```

```
        <members>HideIntroductionLeadEmail.genAiPromptTemplateActivation</members>
    </types>
    <version>60.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

GlobalPicklist

Represents a global picklist, or the set of shared picklist values that custom picklist fields can use. In contrast, the custom picklist fields that are based on a global picklist are of type CustomValue. This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

GlobalPicklist components have the suffix `.globalPicklist` and are stored in the `globalPicklist` folder.

Version

GlobalPicklist components are available in API version 37.0 only. In API version 38.0 and later, GlobalPicklist is replaced by the GlobalValueSet type.

Fields

Field Name	Field Type	Description
`description`	string	It's useful to state the global picklist's purpose, and which objects it's intended for. Limit: 255 characters.
`globalPicklistValues`	GlobalPicklistValue[]	Requires at least one value. The list of values, or "picklist value set," that's defined for a global picklist. The picklist value set is inherited by any custom picklist field that's based on that global picklist. Each value is of type GlobalPicklistValue. A global picklist can have up to 1,000 total values, including inactive values.
`masterLabel`	string	Required. A global picklist's name, which is defined when the global picklist is created. Appears as Label in the user interface.
`sorted`	string	Indicates whether a global picklist's value set is sorted in alphabetical order. By default this value is `false`.

Declarative Metadata Sample Definition

This Territories.globalPicklist is an example of a GlobalPicklist component.

```
<?xml version="1.0" encoding="UTF-8"?>
<GlobalPicklist xmlns="http://soap.sforce.com/2006/04/metadata">
    <description>Updated:This is a basic global picklist</description>
    <globalPicklistValues>
        <fullName>Northwest</fullName>
        <default>false</default>
    </globalPicklistValues>
    <globalPicklistValues>
        <fullName>Northeast</fullName>
        <default>false</default>
    </globalPicklistValues>
    <globalPicklistValues>
        <fullName>South</fullName>
        <default>true</default>
    </globalPicklistValues>
    <globalPicklistValues>
        <fullName>Southwest</fullName>
        <default>false</default>
        <isActive>false</isActive>
    </globalPicklistValues>
    <masterLabel>Territories</masterLabel>
    <sorted>true</sorted>
</GlobalPicklist>
```

This example `package.xml` references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>Territories</members>
        <name>GlobalPicklist</name>
    </types>
    <version>37.0</version>
</Package>
```

GlobalPicklistValue

Represents the definition of a value used in a global picklist. Custom picklist fields can inherit the picklist value set from a global picklist.

This type extends the Metadata metadata type and inherits its `fullName` field.

Note: GlobalPicklistValue components don't have file suffixes or directories because they're lists of values and not custom fields. For file-based operations they can be accessed through GlobalPicklist (which is in API v37.0 only).

Version

GlobalPicklistValue components are available in API version 37.0 only. In API version 38.0 and later, GlobalPicklistValue is replaced by CustomValue on page 761.

Fields

Field Name	Field Type	Description
`color`	string	The color assigned to the picklist value when it's used in charts on reports and dashboards. The color is in hexadecimal format; for example, #FF6600. If a color is not specified, it's assigned dynamically upon chart generation.
`default`	boolean	Required. Indicates whether this value is the default selection for the global picklist and the custom picklists that share its picklist value set. This field is set to *`true`* by default.
`description`	string	The global picklist value's description. It's useful to include a description for a global picklist value so the reason for creating it can be tracked. Limit: 255 characters.
`isActive`	boolean	Indicates whether this value is currently active or inactive. The default value is *`true`*. Users can select only active values from a picklist. An API retrieve operation for global picklist values returns all active and inactive values in the picklist. (Meanwhile, retrieving the values of a non-global, unrestricted picklist returns only the active values.)

PicklistValue

This metadata type defines a value in the picklist and specifies whether this value is the default value. This type extends the GlobalPicklistValue metadata type and inherits all its fields. In API version 36.0 and earlier, PicklistValue extends the Metadata type and inherits its `fullName` field.

Note the following when working with picklist values:

- When you retrieve a standard object, all picklist values are retrieved, not just the customized picklist values.
- When you deploy changes to standard picklist fields, picklist values are added as needed.
- To deactivate a global picklist value, you can invoke an `update()` call on GlobalPicklist with the value omitted, or with the value's `isActive` field set to `false`. Or, you can invoke an `update()` call directly on GlobalPicklistValue with the `isActive` field set to `false`.
- If picklist values are missing from a component definition, they get deactivated when deployed. Deactivation occurs for picklist values of both standard and custom fields.

Field Name	Field Type	Description
`allowEmail`	boolean	Indicates whether this value lets users email a quote PDF (`true`), or not (`false`). This field is only relevant for the `Status` field in quotes. This field is available in API version 18.0 and later.
`closed`	boolean	Indicates whether this value is associated with a closed status (`true`), or not (`false`). This field is only relevant for the standard `Status` field in cases and tasks. This field is available in API version 16.0 and up to version 36.0. In version 37.0, this field is in GlobalPicklistValue.

Field Name	Field Type	Description
`controllingFieldValues`	string[]	A list of values in the controlling field that are linked to this picklist value. The controlling field can be a checkbox or a picklist. This field is available in API version 14.0 and later. The values in the list depend on the field type: • `Checkbox`: `checked` or `unchecked`. • `Picklist`: The `fullname` of the picklist value in the controlling field.
`converted`	boolean	Indicates whether this value is associated with a converted status (`true`), or not (`false`). This field is relevant for only the standard `Lead Status` field in leads. Your organization can set its own guidelines for determining when a lead is qualified, but typically, you want to convert a lead as soon as it becomes a real opportunity that you want to forecast. For more information, see "Convert Qualified Leads" in the Salesforce online help. This field is available in API version 16.0 and later.
`cssExposed`	boolean	Indicates whether this value is available in your Self-Service Portal (`true`), or not (`false`). This field is only relevant for the standard `Case Reason` field in cases. Self-Service provides an online support channel for your customers - allowing them to resolve their inquiries without contacting a customer service representative. For more information about Self-Service, see "Setting Up Your Self-Service Portal" in the Salesforce online help. Note: Starting with Spring '12, the Self-Service portal isn't available for new Salesforce orgs. Existing orgs continue to have access to the Self-Service portal. This field is available in API version 16.0 and later.
`forecastCategory`	ForecastCategories (enumeration of type string)	Indicates whether this value is associated with a forecast category (`true`), or not (`false`). This field is only relevant for the standard `Stage` field in opportunities. • Omitted • Pipeline • BestCase • Forecast • Closed This field is available in API version 16.0 and later.
`highPriority`	boolean	Indicates whether this value is a high priority item (`true`), or not (`false`). This field is only relevant for the standard `Priority` field in tasks. For more information about tasks, see "Start Using Tasks" in the Salesforce online help. This field is available in API version 16.0 and later.

Field Name	Field Type	Description
`probability`	int	Indicates whether this value is a probability percentage (`true`), or not (`false`). This field is only relevant for the standard `Stage` field in opportunities. This field is available in API version 16.0 and later.
`reverseRole`	string	A picklist value corresponding to a reverse role name for a partner. If the role is "subcontractor", then the reverse role might be "general contractor". Assigning a partner role to an account in Salesforce creates a reverse partner relationship so that both accounts list the other as a partner. This field is only relevant for partner roles. For more information, see "Partner Fields" in the Salesforce online help. This field is available in API version 18.0 and later.
`reviewed`	boolean	Indicates whether this value is associated with a reviewed status (`true`), or not (`false`). This field is only relevant for the standard `Status` field in solutions. For more information about opportunities, see "Creating Solutions" in the Salesforce online help. This field is available in API version 16.0 and later.
`won`	boolean	Indicates whether this value is associated with a closed or won status (`true`), or not (`false`). This field is only relevant for the standard `Stage` field in opportunities. This field is available in API version 16.0 and later.

Declarative Metadata Sample Definition

For an example of GlobalPicklistValue components with a `package.xml` that references them, see GlobalPicklist.

Wildcard Support in the Manifest File

This metadata type doesn't support the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

GlobalValueSet

Represents the metadata for a global picklist value set, which is the set of shared values that custom picklist fields can use. A global value set isn't a field itself. In contrast, the custom picklist fields that are based on a global picklist are of type ValueSet. This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

GlobalValueSet components have the suffix `.globalValueSet` and are stored in the `globalValueSets` folder.

Version

GlobalValueSet components are available in API version 38.0 and later. In API version 37.0, this component is the GlobalPicklist type.

Fields

Field Name	Field Type	Description
`customValue`	CustomValue[]	Requires at least one value. The list of values, or "global value set," that's defined for a global picklist. The global value set is inherited by any custom picklist field that uses that value set. Each value is of type customValue. A global value set can have up to 1,000 total values, including inactive values.
`description`	string	It's useful to state the global value set's purpose and which objects it's intended for. Limit: 255 characters.
`masterLabel`	string	Required. A global value set's name, which is defined when the global value set is created. Appears as Label in the user interface.
`sorted`	boolean	Required. Indicates whether a global value set is sorted in alphabetical order. By default this value is `false`.

Declarative Metadata Sample Definition

This UpsellGlobal.globalValueSet is an example of a GlobalValueSet component.

```
<?xml version="1.0" encoding="UTF-8"?>
<GlobalValueSet xmlns="http://soap.sforce.com/2006/04/metadata">
   <description>Updated:This is a basic global value set.</description>
   <masterLabel>UpsellGlobal</masterLabel>
   <customValue>
      <fullName>Maybe</fullName>
      <default>false</default>
      <label>Maybe</label>
   </customValue>
   <customValue>
      <fullName>No</fullName>
      <default>false</default>
      <label>No</label>
   </customValue>
   <customValue>
      <fullName>Yes</fullName>
      <default>false</default>
      <label>Yes</label>
   </customValue>
   <sorted>false</sorted>
</GlobalValueSet>
```

This example `package.xml` references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
```

```
    <types>
        <members>UpsellGlobal</members>
        <name>GlobalValueSet</name>
    </types>
    <version>40.0</version>
</Package>
```

Any global value set created in API version 57.0 or later automatically has the `__gvs` suffix appended to the developer name. When you make any CRUD-based call with the GlobalValueSet type, you must append the suffix to the fullName field when you reference the type.

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

GlobalValueSetTranslation

Contains details for a global value set translation. Global value sets are lists of values that can be shared by multiple custom picklist fields, optionally across objects. This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

GlobalValueSetTranslation components have the suffix `.globalValueSetTranslation` and are stored in the `globalValueSetTranslations` folder.

Translations are stored in a file with a format of `ValueSetName-lang.globalValueSetTranslation`, where *`ValueSetName`* is the global value set's name, and *`lang`* is the translation language.

Version

GlobalValueSetTranslation components are available in API version 38.0 and later.

Fields

Field	Field Type	Description
`valueTranslation`	ValueTranslation[]	The translated name of a value in a translated global value set. Each `valueTranslation` is paired with a `masterLabel`, which is the original (untranslated) name of the value.

ValueTranslation

The original value name and the translated value name in a translated global value set.

Field	Field Type	Description
`masterLabel`	string	Required. The original (untranslated) name of a value in a global value set. Each `valueTranslation` has a `masterLabel` paired with its `translation`.
`translation`	string	The translated name of a value in a translated global value set.

Declarative Metadata Sample Definition

This example shows a GlobalValueSetTranslation component. When a value isn't translated, its translation becomes a comment that's paired with its masterLabel.

```
<?xml version="1.0" encoding="UTF-8"?>
<GlobalValueSetTranslation xmlns="http://soap.sforce.com/2006/04/metadata">
    <valueTranslation>
        <masterLabel>Three</masterLabel>
        <translation>Trois</translation>
    </valueTranslation>
    <valueTranslation>
        <masterLabel>Four</masterLabel>
        <translation>Quatre</translation>
    </valueTranslation>
    <valueTranslation>
        <masterLabel>Five</masterLabel>
        <translation><!-- Five --></translation>
    </valueTranslation>
</GlobalValueSetTranslation>
```

This example is a `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
        <types>
        <members>Numbers-fr</members>
        <name>GlobalValueSetTranslation</name>
    </types>
    <version>38.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SEE ALSO:

Translations

Group

Represents a set of public groups, which can have users, roles, and other groups.

Declarative Metadata File Suffix and Directory Location

The file suffix for group components is `.group` and components are stored in the `groups` directory of the corresponding package directory.

Version

Group components are available in API version 24.0 and later.

Special Access Rules

As of Spring '20 and later, only authenticated internal and external users can access this type.

Fields

Note: Members of the public group are not migrated when you deploy the group type.

This metadata type represents the valid values that define a group:

Field Name	Field Type	Description
`description`	string	The description for the group. Available in API version 62.0 and later.
`doesIncludeBosses`	boolean	Indicates whether records shared with users in this group are also shared with users higher in the role hierarchy (`true`) or not (`false`). This field is only available for public groups. This field corresponds to the Grant Access Using Hierarchies checkbox in Setup.
`fullName`	string	The unique identifier for API access. The `fullName` can contain only underscores and alphanumeric characters. It must be unique, begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. This field is inherited from the Metadata component. Corresponds to **Group Name** in the user interface.
`name`	string	Required. The name of the group. Corresponds to **Label** in the user interface.

Declarative Metadata Sample Definition

The following is the definition of a group.

```
<?xml version="1.0" encoding="UTF-8"?>
<Group xmlns="http://soap.sforce.com/2006/04/metadata">
    <doesIncludeBosses>true</doesIncludeBosses>
    <fullName>admin</fullName>
```

```
    <name>test</name>
</Group>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

HomePageComponent

Represents the metadata associated with a home page component. You can customize the Home tab in Salesforce Classic to include components such as sidebar links, a company logo, a dashboard snapshot, or custom components that you create. Use to create, update, or delete home page component definitions.

For more information, see "Salesforce Classic Home Tab Page Layouts" in the Salesforce Help.

This type extends the Metadata metadata type and inherits its `fullName` field.

Declarative Metadata File Suffix and Directory Location

The file suffix for home page components is `.homePageComponent` and components are stored in the `homepagecomponents` directory of the corresponding package directory.

Version

Home page components are available in API version 12.0 and later.

HomePageComponent

This metadata type represents the valid values that define a home page component:

Field Name	Field Type	Description
`body`	string	The text body inside the HTML page component.
`fullName`	string	The name can only contain characters, letters, and the underscore (_) character. The name must start with a letter, and can't end with an underscore or contain two consecutive underscore characters. Inherited from the Metadata component, this field isn't defined in the WSDL for this component. It must be specified when creating, updating, or deleting. See create() to see an example of this field specified for a call.
`height`	int	Required for Visualforce Area components. Indicates the height (in pixels) of the component. This field is available in API version 31.0 and later.

Field Name	Field Type	Description
`links`	string[]	If the `pageComponentType` is `links`, then zero or more names of custom page links can be specified. • `ObjectWebLink` • `CustomPageWebLink`
`page`	string	This field is only available for Visualforce Area components and indicates the API name of the Visualforce page that is associated with the component. This field is available in API version 31.0 and later.
`pageComponentType`	PageComponentType (enumeration of type string)	Required. Valid values are: • `links` • `htmlArea` • `imageOrNote` • `visualforcePage` (This value is available in API version 31.0 and later.)
`showLabel`	boolean	This field is only available for Visualforce Area components and specifies whether the component displays with a label (`true`) or not (`false`). This field is available in API version 31.0 and later.
`showScrollbars`	boolean	This field is only available for Visualforce Area components and specifies whether the component displays with scrollbars (`true`) or not (`false`). This field is available in API version 31.0 and later.
`width`	PageComponentWidth (enumeration of type string)	This field is only available for HTML and Visualforce Area components, and indicates whether it's a narrow or wide home page component. Valid values are: • `narrowComponents` • `wideComponents`

Declarative Metadata Sample Definition

The following is the definition of a home page component. See Declarative Metadata Sample Definition and Declarative Metadata Sample Definition for related samples.

```
<?xml version="1.0" encoding="UTF-8"?>
<HomePageComponent xmlns="http://soap.sforce.com/2006/04/metadata">
  <height>200</height>
  <page>MyVisualforcePage</page>
  <pageComponentType>visualforcePage</pageComponentType>
  <showLabel>true</showLabel>
  <showScrollbars>true</showScrollbars>
```

```
  <width>wideComponents</width>
</HomePageComponent>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SEE ALSO:

- HomePageLayout
- WebLink

HomePageLayout

Represents the metadata associated with a home page layout. You can customize home page layouts and assign the layouts to users based on their user profile.

File Suffix and Directory Location

Home page layouts are stored in the `homePageLayouts` directory of the corresponding package directory. The extension is `.homePageLayout`.

Version

Home page components are available in API version 12.0 and later. This type extends the Metadata metadata type and inherits its `fullName` field.

Fields

This metadata type represents the valid values that define a home page layout:

Field Name	Field Type	Description
`fullName`	string	The name can only contain characters, letters, and the underscore (_) character. The name must start with a letter, and can't end with an underscore or contain two consecutive underscore characters. Inherited from the Metadata component, this field isn't defined in the WSDL for this component. It must be specified when creating, updating, or deleting. See create() to see an example of this field specified for a call.
`narrowComponents`	string[]	The list of elements in the narrow column on the left side of the home page.
`wideComponents`	string[]	The list of elements in the wide column on the right side of the home page.

Declarative Metadata Sample Definition

The following is the definition of a home page layout. See Declarative Metadata Sample Definition on page 1298 and Declarative Metadata Sample Definition on page 729 for related samples.

```
<?xml version="1.0" encoding="UTF-8"?>
<HomePageLayout xmlns="http://soap.sforce.com/2006/04/metadata">
    <narrowComponents>google</narrowComponents>
</HomePageLayout>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SEE ALSO:

- HomePageComponent
- WebLink

IdentityVerificationProcDef

Represents the definition of the identity verification process.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

IdentityVerificationProcDef components have the suffix `.IdentityVerificationProcDef` and are stored in the `IdentityVerificationProcDefs` folder.

Version

IdentityVerificationProcDef components are available in API version 54.0 and later.

Special Access Rules

The Health Cloud permission set license is required to use this metadata type.

Fields

Field Name	Description
`identityVerificationProcDtls`	**Field Type** IdentityVerificationProcDtl[] **Description** A list of Identity Verification Process Detail elements.
`masterLabel`	**Field Type** string **Description** Required. The label of the Identity Verification Process Definition record.
`searchLayoutType`	**Field Type** IdentityVerificationSearchLayoutType (enumeration of type string) **Description** Required. The display layout of the search component. Valid values are: • `Stack` • `Tab`

IdentityVerificationProcDtl

Represents the verification-related details such as search criteria, verification criteria, or the custom apex class.

Field Name	Description
`apexClass`	**Field Type** string **Description** The Apex class that is used to search and verify data in an external system.
`dataSourceType`	**Field Type** IdentityVerificationDataSourceType (enumeration of type string) **Description** Required. The source type of the data. Valid values are:

Field Name	Description
	• `External` • `Salesforce`
`developerName`	**Field Type** string **Description** Required. The developer name of Identity verification process detail. Can contain only underscores and alphanumeric characters and must be unique in your org. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
`displayRecordFieldName`	**Field Type** string **Description** The name of the field that contains information about the record that's shown to the user after identity verification is successful. Available in API version 58.0 and later.
`identityVerificationProcFlds`	**Field Type** IdentityVerificationProcFld[] **Description** A list of Identity Verification Process Field elements.
`isActive`	**Field Type** boolean **Description** Indicates whether the record is active (`true`) or not (`false`).
`isRetryAllowedAfterLimit`	**Field Type** boolean **Description** For internal use only.
`linkedIdVerfProcessDet`	**Field Type** string **Description** The record containing the details of the linked identity verification process. Available in API version 58.0 and later.
`masterLabel`	**Field Type** string

Field Name	Description
	Description Required. The label of the Identity Verification Process Detail record.
`objectName`	**Field Type** string **Description** The name of the object on which the search is performed and data is verified.
`optionalVerifiersMinVerfCount`	**Field Type** int **Description** The number of optional verifiers that must be checked.
`retryLimit`	**Field Type** int **Description** For internal use only.
`searchFilter`	**Field Type** string **Description** A comma-separated list of predefined filter conditions that are used to refine the scope of the search.
`searchRecordUniqueIdField`	**Field Type** string **Description** The field storing the unique identifier of a record displayed in the search results.
`searchResultSortBy`	**Field Type** string **Description** The values that are used to sort the search results.
`searchSequenceNumber`	**Field Type** int **Description** Required. The sequence in which the search is performed and the search result is displayed.

Field Name	Description
`searchType`	**Field Type** IdentityVerificationSearchType (enumeration of type string) **Description** Required. The type of search being performed. Valid values are: • `Object-Based` • `Text-Based`

IdentityVerificationProcFld

Represents a set of fields necessary to configure the questions that CCA asks the caller before providing them with the information they need.

Field Name	Description
`customFieldLabel`	**Field Type** string **Description** The custom label for the field that contains the verification data.
`dataSourceType`	**Field Type** IdentityVerificationProcFldDataSourceType (enumeration of type string) **Description** Required. The source type of the data. Valid values are: • `External` • `Salesforce`
`developerName`	**Field Type** string **Description** Required. The developer name of Identity Verification Process Field. Can contain only underscores and alphanumeric characters and must be unique in your org. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. Available in API version 58.0 and later.

Field Name	Description
`fieldDataType`	**Field Type** IdentityVerificationProcFldFieldDataType (enumeration of type string) **Description** The data type of the field in the external data source that's defined in the identity verification process detail. Available in API version 58.0 and later. Valid values are: • `address` • `checkbox` • `currency` • `dateonly` • `datetime` • `email` • `number` • `other` • `percent` • `phone` • `picklist` • `reference` • `text` • `timeonly` • `url`
`fieldName`	**Field Type** string **Description** Required. The label of the field that contains the verification data based on the selected field type. Available in API version 58.0 and later.
`fieldType`	**Field Type** IdentityVerificationProcFldFieldType (enumeration of type string) **Description** Required. Indicates the type of field. Valid values are: • `additionalResultField` • `optionalVerifier` • `requiredVerifier`

Field Name	Description
	• `resultField` • `searchField` • `searchFilter`
`fieldValueFormula`	**Field Type** string **Description** Stores the formula that is applied to the field value.
`isActive`	**Field Type** boolean **Description** Indicates whether the record is active (`true`) or not (`false`).
`isManualInput`	**Field Type** boolean **Description** Indicates whether the user can manually enter the identity verification details (`true`) or not (`false`). The default value of this field is `false`. Available in API version 58.0 and later.
`masterLabel`	**Field Type** string **Description** Required. A user-friendly name for Identity Verification Process Field.
`sequenceNumber`	**Field Type** int **Description** Required. The sequence number of the field.

Declarative Metadata Sample Definition

This is an example of an IdentityVerificationProcDef component.

```
<?xml version="1.0" encoding="UTF-8"?>
<IdentityVerificationProcDef xmlns="http://soap.sforce.com/2006/04/metadata">
   <identityVerificationProcDtls>
```

```
    <fullName>Sample93AccountSearch</fullName>
    <dataSourceType>Salesforce</dataSourceType>
    <developerName>Sample93AccountSearch</developerName>
    <identityVerificationProcFlds>
        <fullName>Sample93AccountName</fullName>
        <dataSourceType>Salesforce</dataSourceType>
        <developerName>Sample93AccountName</developerName>
        <fieldName>Name</fieldName>
        <fieldType>requiredVerifier</fieldType>
        <isActive>false</isActive>
        <masterLabel>Sample93 Account Name</masterLabel>
        <fieldValueFormula>abcd</fieldValueFormula>
        <customFieldLabel>Name</customFieldLabel>
        <sequenceNumber>1</sequenceNumber>
        <isManualInput>false</isManualInput>
    </identityVerificationProcFlds>
    <identityVerificationProcFlds>
        <fullName>Sample93Phone</fullName>
        <dataSourceType>Salesforce</dataSourceType>
        <developerName>Sample93Phone</developerName>
        <fieldName>phone</fieldName>
        <fieldType>optionalVerifier</fieldType>
        <isActive>false</isActive>
        <masterLabel>Sample93 Phone</masterLabel>
        <sequenceNumber>93</sequenceNumber>
        <isManualInput>false</isManualInput>
    </identityVerificationProcFlds>
    <identityVerificationProcFlds>
        <fullName>Sample93PostalCode</fullName>
        <dataSourceType>Salesforce</dataSourceType>
        <developerName>Sample93PostalCode</developerName>
        <fieldName>BillingPostalCode</fieldName>
        <fieldType>optionalVerifier</fieldType>
        <isActive>true</isActive>
        <masterLabel>Sample93 Postal Code</masterLabel>
        <sequenceNumber>4</sequenceNumber>
        <isManualInput>false</isManualInput>
    </identityVerificationProcFlds>
    <identityVerificationProcFlds>
        <fullName>Sample93Account</fullName>
        <dataSourceType>Salesforce</dataSourceType>
        <developerName>Sample93Account</developerName>
        <fieldName>Name</fieldName>
        <fieldType>resultField</fieldType>
        <isActive>false</isActive>
        <masterLabel>Sample93 Account</masterLabel>
        <sequenceNumber>1</sequenceNumber>
        <isManualInput>false</isManualInput>
    </identityVerificationProcFlds>
    <isActive>true</isActive>
    <masterLabel>Sample93 Account Search</masterLabel>
    <objectName>Account</objectName>
    <searchRecordUniqueIdField>Id</searchRecordUniqueIdField>
    <searchSequenceNumber>1</searchSequenceNumber>
```

```
      <searchType>Text-Based</searchType>
      <searchResultSortBy>Name</searchResultSortBy>
      <optionalVerifiersMinVerfCount>1</optionalVerifiersMinVerfCount>
      <isRetryAllowedAfterLimit>false</isRetryAllowedAfterLimit>
      <retryLimit>5</retryLimit>
      <searchFilter></searchFilter>
      <displayRecordFieldName>LastModifiedById</displayRecordFieldName>
   </identityVerificationProcDtls>
   <masterLabel>Sample93 Verification Flow</masterLabel>
   <searchLayoutType>Tab</searchLayoutType>
</IdentityVerificationProcDef>
```

This is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>IdentityVerificationProcDef</name>
    </types>
    <version>54.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

IdentityVerificationProcDtl

Represents the search functionality configuration and the minimum number of optional verifiers for identity verification. This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

IdentityVerificationProcDtl components have the suffix `.IdentityVerificationProcDtl` and are stored in the `IdentityVerificationProcDtls` folder.

Version

IdentityVerificationProcDtl components are available in API version 54.0 and later.

Special Access Rules

The Health Cloud permission set license is required to use this metadata type.

Fields

Field Name	Description
`apexClass`	**Field Type** string **Description** Reserved for future use.
`dataSourceType`	**Field Type** IdentityVerificationDataSourceType (enumeration of type string) **Description** Required. The source type of the data. Valid values are: • `External`—Reserved for future use. • `Salesforce`
`identityVerificationProcFlds`	**Field Type** IdentityVerificationProcFld[] **Description** A list of Identity Verification Process Field elements.
`isActive`	**Field Type** boolean **Description** Indicates whether the record is active (`true`) or not (`false`). The default value is `false`.
`masterLabel`	**Field Type** string **Description** Required. The label of the Identity Verification Process Detail record.
`objectName`	**Field Type** string **Description** The name of the object on which the search is performed and data is verified.
`optionalVerifiersMinVerfCount`	**Field Type** int

Field Name	Description
	Description The minimum number of optional verifiers that must be checked.
`searchFilter`	**Field Type** string **Description** Conditions on which to filter the search results. For example, if you want to perform the search only on Person Account records, enter `isPersonAccount = true`.
`searchRecordUniqueIdField`	**Field Type** string **Description** The field that stores the unique identifier of the records that are displayed in the search results.
`searchResultSortBy`	**Field Type** string **Description** The values that are used to sort the search results. For example, if you want to sort the results by policy date, enter `PolicyDate__c Desc`.
`searchSequenceNumber`	**Field Type** int **Description** Required. Enter 1 as the search sequence number. Note: In API version 54.0 and later, this field is reserved for future use, and the value you enter doesn't affect sequencing.
`searchType`	**Field Type** IdentityVerificationSearchType (enumeration of type string) **Description** Required. The type of search being performed. Valid values are: • `Object-Based`—Reserved for future use. • `Text-Based`

IdentityVerificationProcFld

Represents a set of fields necessary to configure the questions that CCA asks the caller before providing them with the information they need.

Field Name	Description
`customFieldLabel`	**Field Type** string **Description** The custom label for the field that contains the verification data.
`dataSourceType`	**Field Type** IdentityVerificationProcFldDataSourceType (enumeration of type string) **Description** Required. The source type of the data. Valid values are: • `External` • `Salesforce`
`developerName`	**Field Type** string **Description** Required. The developer name of Identity Verification Process Field. Can contain only underscores and alphanumeric characters and must be unique in your org. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. Available in API version 58.0 and later.
`fieldDataType`	**Field Type** IdentityVerificationProcFldFieldDataType (enumeration of type string) **Description** The data type of the field in the external data source that's defined in the identity verification process detail. Available in API version 58.0 and later. Valid values are: • `address` • `checkbox` • `currency` • `dateonly` • `datetime` • `email`

Field Name	Description
	• number • other • percent • phone • picklist • reference • text • timeonly • url
fieldName	**Field Type** string **Description** Required. The label of the field that contains the verification data based on the selected field type. Available in API version 58.0 and later.
fieldType	**Field Type** IdentityVerificationProcFldFieldType (enumeration of type string) **Description** Required. Indicates the type of field. Valid values are: • additionalResultField • optionalVerifier • requiredVerifier • resultField • searchField • searchFilter
fieldValueFormula	**Field Type** string **Description** Stores the formula that is applied to the field value.
isActive	**Field Type** boolean **Description** Indicates whether the record is active (true) or not (false).

Field Name	Description
isManualInput	**Field Type** boolean **Description** Indicates whether the user can manually enter the identity verification details (`true`) or not (`false`). The default value of this field is `false`. Available in API version 58.0 and later.
masterLabel	**Field Type** string **Description** Required. A user-friendly name for Identity Verification Process Field.
sequenceNumber	**Field Type** int **Description** Required. The sequence number of the field.

Declarative Metadata Sample Definition

The following is an example of an identityVerificationProcDtl component.

```
<?xml version="1.0" encoding="UTF-8"?>
<IdentityVerificationProcDtl xmlns="http://soap.sforce.com/2006/04/metadata">
    <dataSourceType>Salesforce</dataSourceType>
    <isActive>true</isActive>    <developerName>Sample93AccountSearch</developerName>
    <identityVerificationProcFlds>
       <fullName>Sample93AccountName</fullName>
       <dataSourceType>Salesforce</dataSourceType>
       <developerName>Sample93AccountName</developerName>
       <fieldName>Name</fieldName>
       <fieldType>requiredVerifier</fieldType>
       <isActive>false</isActive>
       <masterLabel>Sample93 Account Name</masterLabel>
       <fieldValueFormula>abcd</fieldValueFormula>
       <customFieldLabel>Name</customFieldLabel>
       <sequenceNumber>1</sequenceNumber>
       <isManualInput>false</isManualInput>
    </identityVerificationProcFlds>
    <identityVerificationProcFlds>
       <fullName>Sample93Phone</fullName>
```

```
        <dataSourceType>Salesforce</dataSourceType>
        <developerName>Sample93Phone</developerName>
        <fieldName>phone</fieldName>
        <fieldType>optionalVerifier</fieldType>
        <isActive>false</isActive>
        <masterLabel>Sample93 Phone</masterLabel>
        <sequenceNumber>93</sequenceNumber>
        <isManualInput>false</isManualInput>
    </identityVerificationProcFlds>
    <masterLabel>detail1</masterLabel>
     <fullName>detail1</fullName>
    <objectName>Account</objectName>
    <optionalVerifiersMinVerfCount>11</optionalVerifiersMinVerfCount>
    <searchFilter>asd</searchFilter>
    <searchRecordUniqueIdField>Id</searchRecordUniqueIdField>
    <searchResultSortBy>asd</searchResultSortBy>
    <searchSequenceNumber>1</searchSequenceNumber>
    <searchType>Text-Based</searchType>
</IdentityVerificationProcDtl>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>IdentityVerificationProcDtl</name>
    </types>
    <version>54.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

IdentityVerificationProcFld

Represents the search and verification fields used in identity verification. This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

IdentityVerificationProcFld components have the suffix `.IdentityVerificationProcFld` and are stored in the `IdentityVerificationProcFlds` folder.

Version

IdentityVerificationProcFld components are available in API version 54.0 and later.

Special Access Rules

The Health Cloud permission set license is required to use this metadata type.

Fields

Field Name	Description
`customFieldLabel`	**Field Type** string **Description** The custom label for the field that contains the verification data. Note: Translation of custom field labels isn't supported in API version 54.0.
`dataSourceType`	**Field Type** IdentityVerificationProcFldDataSourceType (enumeration of type string) **Description** Required. The source type of the data. Valid values are: • `External` Note: An external data source isn't supported in API version 54.0. • `Salesforce`
`fieldDataType`	**Type** picklist **Properties** Filter, Group, Nillable, Restricted picklist, Sort **Description** The type of data stored in an external data source field. Possible values are: • `address` • `checkbox` • `currency` • `dateonly` • `datetime`

Field Name	Description
	• email • number • other • percent • phone • picklist • reference • text • timeonly • url
fieldName	**Field Type** string **Description** Required. The label of the field that contains the verification data based on the selected field type.
fieldType	**Field Type** IdentityVerificationProcFldFieldType (enumeration of type string) **Description** Required. Indicates the type of field. Possible values are: • additionalResultField—Fetches data as part of the search query, but the data isn't displayed in search results. Use this value if, for example, you want to fetch the policy number and the age of the policy owner as a result of the search, but the agent isn't supposed to see this data. You can write custom logic to process this additional data. • optionalVerifier—Optional verifier. • requiredVerifier—Required verifier. • resultField—Displays field type in search results. Use this value if, for example, when an agent searches for a caller, you'd like the search results to include the account name, phone number, and email ID. • searchField—Reserved for future use. • searchFilter—A comma-separated list of predefined filter conditions that are used to refine the scope of the search.
fieldValueFormula	**Field Type** string

Field Name	Description
	Description Reserved for future use.
isActive	**Field Type** boolean **Description** Indicates whether the record is active (`true`) or not (`false`). The default value is `false`.
isManualInput	**Field Type** boolean **Description** Indicates whether the user can manually enter the identity verification details (`true`) or not (`false`). The default value is `false`. This field is available in API version 58.0 and later.
masterLabel	**Field Type** string **Description** Required. The label of the Identity Verification Process Field record.
sequenceNumber	**Field Type** int **Description** Required. The sequence number of the field.

Declarative Metadata Sample Definition

The following is an example of an IdentityVerificationProcFld component.

```
<?xml version="1.0" encoding="UTF-8"?>
<IdentityVerificationProcFld xmlns="http://soap.sforce.com/2006/04/metadata"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <customFieldLabel>field1</customFieldLabel>
    <dataSourceType>External</dataSourceType>
    <fieldName>sasa</fieldName>
    <fieldType>requiredVerifier</fieldType>
```

```
     <fullName>field1</fullName>

    <isActive>false</isActive>
    <masterLabel>field1</masterLabel>
    <sequenceNumber>1</sequenceNumber>
</IdentityVerificationProcFld>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>IdentityVerificationProcFld</name>
    </types>
    <version>54.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

InboundCertificate

Represents a mutual authentication certificate that is imported to your Salesforce org.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

InboundCertificate components have the suffix `.inboundCertificate` and are stored in the `inboundCertificates` folder.

Special Access Rules

InboundCertificate is available when the MutualAuthentication permission is enabled in your org.

Version

InboundCertificate components are available in API version 49.0 and later.

Fields

This metadata type contains the following fields:

Field Name	Field Type	Description
`expirationDate`	date	Required. The date on which the certificate expires.
`issuer`	string	Required. The certificate's issuer.
`masterLabel`	string	Required. A friendly name that you create for the certificate. Limited to 64 characters.
`serialId`	string	Required. The serial number for the certificate.

Declarative Metadata Sample Definition

The following is an example of an InboundCertificate component.

```
<InboundCertificate xmlns="http://soap.sforce.com/2006/04/metadata">
    <expirationDate>2021-02-04</expirationDate>
    <issuer>C=USA,ST=CA,L=San
Francisco,O=Salesforce.com,OU=00Dxx0000006Jm7,CN=newTestCert</issuer>
    <masterLabel>TestMutualAuthCert2</masterLabel>
    <serialId>291613202525313237574705460716 24</serialId>
</InboundCertificate>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>InboundCertificate</name>
    </types>
    <version>49.0</version>
</Package>
```

Usage

To prevent simple impersonation from compromising security, you can require clients and servers to prove their identity to each other with a mutual authentication certificate.

InboundNetworkConnection

Represents a private connection between a third-party data service and a Salesforce org. The connection is inbound because the callouts are coming *into* Salesforce.This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

InboundNetworkConnection components have the suffix `.inboundNetworkConnection` and are stored in the `inboundNetworkConnections` folder.

Version

InboundNetworkConnection components are available in API version 49.0 and later.

Fields

Field Name	Field Type	Description
`connectionType`	ExternalConnectionType (enumeration of type string)	Required. Specifies the cloud provider of the connection. • `AwsPrivateLink` • `DataCloudPrivateConnection` (Reserved for internal use)
`description`	string	Required. A description of the connection. Maximum of 255 characters.
`inboundNetworkConnProperties`	InboundNetworkConnProperty	Name-value pairs that describe the properties of the inbound network connection. Specify a name-value pair for each of the properties.
`isActive`	boolean	Required. Specifies whether the connection is active (`true`) or not(`false`). The default value is `false`.
`label`	string	Required. A user-friendly label for the connection.
`status`	ExternalConnectionStatus (enumeration of type string)	Required. Connection status. The connection is initially Unprovisioned and moves through the other states automatically after an admin performs a Provision, Sync, or Teardown action. The valid values are: • `Unprovisioned` • `Allocating` • `PendingAcceptance` • `PendingActivation` • `RejectedRemotely` • `DeletedRemotely` • `TeardownInProgress` • `Ready`

InboundNetworkConnProperty

Represents a name-value pair that describes the properties of the inbound network connection.

Field Name	Field Type	Description
`propertyName`	InboundConnPropertyName (enumeration of type string)	Required. The name of a property used to establish an InboundNetworkConnection. Valid values are: • `AwsVpcEndpointId`—The unique endpoint ID for connections to an AWS Virtual Private Cloud (VPC). The value is read-only when the `status` is `Ready`. • `Region`—The region in which the VPC is hosted.

Field Name	Field Type	Description
		• `SourceIpRanges`—The ranges of source IP address allocated to this inbound connection by the Salesforce-managed VPC in your cloud provider.
`propertyValue`	string	Required. The value of InboundConnPropertyName. An example of the `propertyValue` of `Region` is `us-west-2`. The `propertyValue` of `SourceIpRanges` is a JSON string that lists the start and end IP address for each range. This example shows two IP address ranges. `[ { "startIp":"10.10.10.0", "endIp":"10.10.10.3" }, { "startIp":"100.100.100.0", "endIp":"100.100.100.15" } ]`

Declarative Metadata Sample Definition

The following sample definition has the suffix `.inboundNetworkConnection`.

```
<?xml version="1.0" encoding="UTF-8"?>
<InboundNetworkConnection xmlns="http://soap.sforce.com/2006/04/metadata">
    <connectionType>AwsPrivateLink</connectionType>
    <description>This is an Inbound Connection to make API calls into
Salesforce</description>
    <inboundNetworkConnProperties>
        <propertyName>Region</propertyName>
        <propertyValue>us-west-2</propertyValue>
    </inboundNetworkConnProperties>
    <inboundNetworkConnProperties>
        <propertyName>AwsVpcEndpointId</propertyName>
        <propertyValue>vpce-02ccb5fac2bacaceb</propertyValue>
    </inboundNetworkConnProperties>
    <inboundNetworkConnProperties>
        <propertyName>SourceIpRanges</propertyName>
        <propertyValue>[ { "startIp":"10.10.10.0", "endIp":"10.10.10.3" }, {
"startIp":"100.100.100.0", "endIp":"100.100.100.15" } ]</propertyValue>
    </inboundNetworkConnProperties>
    <isActive>true</isActive>
    <label>MyInboundConnection</label>
    <status>Unprovisioned</status>
</InboundNetworkConnection>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <fullName>sampleInboundConnection</fullName>
    <types>
        <members>MyInboundConnection</members>
        <name>InboundNetworkConnection</name>
    </types>
    <version>49.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

IndustriesPricingSettings

Represents the settings for Salesforce Pricing.

Parent Type and Manifest Access

This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all the settings metadata types for the org are accessed using the "Settings" name. See Settings for more details.

File Suffix and Directory Location

`IndustriesPricingSettings` values are stored in the `IndustriesPricing.settings` file in the `settings` folder. The `.settings` files are different from other named components, because there's only one settings file for each settings component.

Version

IndustriesPricingSettings components are available in API version 60.0 and later.

Special Access Rules

This metadata type is available with Salesforce Pricing.

Fields

Field Name	Description
`enableHighAvailability`	**Field Type** boolean

Field Name	Description
	Description Reserved for internal use.
`enableLowestPriceCompliance`	**Field Type** boolean **Description** Indicates whether to provide end buyer visibility on lowest price for 30 days (`true`) or not (`false`). The default value is `false`. Available in API version 62.0 and later.
`enablePricingWaterfall`	**Field Type** boolean **Description** Indicates whether to enable Price Waterfall (`true`) or not (`false`). The default value is `false`. Price Waterfall provides insights that include price breakups and reasons for every step of the pricing process.
`enablePricingWaterfallPersistence`	**Field Type** boolean **Description** Indicates whether to enable Price Waterfall Persistence (`true`) or not (`false`). The default value is `false`. Price Waterfall Persistence stores the process logs that provide insights into the internal pricing processes.
`enableSalesforcePricing`	**Field Type** boolean **Description** Indicates whether to enable Salesforce Pricing (`true`) or not (`false`). The default value is `false`.

Declarative Metadata Sample Definition

This example shows a sample IndustriesPricingSettings component.

```
<IndustriesPricingSettings xmlns="http://soap.sforce.com/2006/04/metadata">
     <enableHighAvailability>true</enableHighAvailability>
     <enableLowestPriceCompliance>true</enableLowestPriceCompliance>
     <enablePricingWaterfall>true</enablePricingWaterfall>
     <enablePricingWaterfallPersistence>true</enablePricingWaterfallPersistence>
     <enableSalesforcePricing>true</enableSalesforcePricing>
</IndustriesPricingSettings>
```

This example shows a sample `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
```

```
    <types>
        <members>IndustriesPricing</members>
        <name>Settings</name>
    </types>
    <version>64.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

IndustriesRatingSettings

Represents the settings for Rate Management.

Parent Type and Manifest Access

This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all the settings metadata types for the org are accessed using the "Settings" name. See Settings for more details.

File Suffix and Directory Location

The `IndustriesRatingSettings` values are stored in the `IndustriesRating.settings` file in the `settings` folder. The `.settings` files are different from other named components, because there's only one settings file for each settings component.

Version

IndustriesRatingSettings components are available in API version 62.0 and later.

Special Access Rules

This metadata type is available with Rate Management.

Fields

Field Name	Description
`enableRating`	**Field Type** boolean **Description** Indicates whether to enable Rate Management (`true`) or not (`false`). The default value is `false`.

Field Name	Description
enableRatingWaterfall	**Field Type** boolean **Description** Indicates whether to enable Rating Waterfall (`true`) or not (`false`). The default value is `false`. Rating Waterfall provides insights into the rating data, which you can synchronize with your rating lookup tables.
enableRatingWaterfallPersistence	**Field Type** boolean **Description** Indicates whether to enable Rating Waterfall Persistence (`true`) or not (`false`). The default value is `false`. Rating Waterfall Persistence stores rating data, which you can use to enhance the internal processes and increase efficiency.

Declarative Metadata Sample Definition

The following is an example of an IndustriesRatingSettings component.

```
<IndustriesRatingSettings xmlns="http://soap.sforce.com/2006/04/metadata">
     <enableRating>true</enableRating>
     <enableRatingWaterfall>true</enableRatingWaterfall>
     <enableRatingWaterfallPersistence>true</enableRatingWaterfallPersistence>
</IndustriesRatingSettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>IndustriesRating</members>
        <name>Settings</name>
    </types>
    <version>64.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

InstalledPackage

Represents a first-generation managed package to be installed or uninstalled. Deploying a newer version of a currently installed package upgrades the package. To install an unlocked or second-generation managed package, use the `sf package install` Salesforce CLI command.

Note: You can't deploy a package along with other metadata types. When you deploy InstalledPackage, it must be the only metadata type specified in the manifest file.

File Suffix and Directory Location

The package is specified in the `installedPackages` directory, in a file named after the package's namespace prefix. The file extension is `.installedPackage`.

Version

InstalledPackage is available in API version 28.0 and later.

Fields

Field Name	Field Type	Description
`activateRSS`	boolean	Required. Determines the state of Remote Site Settings (RSS) and Content Security Policy (CSP) at the time of installing the package and must be set to either of these values. **true** Keep the isActive state of any RSS or CSP in the package. **false** Override the isActive state of any RSS or CSP in the package and set it to `false`. The default value is `false`. Available in API version 43.0 and later.
`password`	string	Specifies the package password.
`securityType`	string	Determines user access for the installed package. Valid values are: • `AdminsOnly` • `AllUsers` The default value is `AllUsers`. Available in API version 57.0 and later.
`versionNumber`	string	Required. The version number of the package. The version number has the format ***`majorNumber.minorNumber.patchNumber`*** (for example, `2.1.3`).

Declarative Metadata Sample Definition

The following example specifies a sample package to be installed or uninstalled.

```
<?xml version="1.0" encoding="UTF-8"?>
        <InstalledPackage xmlns="http://soap.sforce.com/2006/04/metadata">
        <versionNumber>1.0</versionNumber>
        <password>optional_password</password>
```

```
        <securityType>AdminsOnly</securityType>
        <activateRSS>true</activateRSS>
        </InstalledPackage>
```

The `securityType` field is optional. If it's not specified, the default security type is `AllUsers`.

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

IntegrationProviderDef

Represents an integration definition associated with a service process. Stores data for the Industries: Send Apex Async Request and Industries: Send External Async Request invocable actions.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

IntegrationProviderDef components have the suffix `.integrationProviderDefinition` and are stored in the `.integrationProviderDefinition` folder.

Version

IntegrationProviderDef components are available in API version 57.0 and later.

Special Access Rules

Access to the IntegrationProviderDef type requires the AccessToServiceProcess permission.

Fields

Field Name	Description
`active`	**Field Type** boolean **Description** Specifies whether this Integration Definition is active. The default is `false`.
`apexClass`	**Field Type** string

Field Name	Description
	Description The custom Apex class that the related Industries: Send Apex Async Request invocable action invokes. Specify either apexClass or fileBasedApexClass but not both. Applies only if the type is `Apex`.
`description`	**Field Type** string **Description** A meaningful explanation of the Integration Definition.
`developerName`	**Field Type** string **Description** Required. A system name for the Integration Definition.
`externalServiceOperationName`	**Field Type** string **Description** The external service operation that the related Industries: Send External Async Request invocable action invokes. Applies only if the type is `LowCode`.
`externalServiceRegistration`	**Field Type** string **Description** The external service that the related Industries: Send External Async Request invocable action invokes. Applies only if the type is `LowCode`.
`fileBasedApexClass`	**Field Type** string **Description** The Salesforce-provided Apex class that the related Industries: Send Apex Async Request invocable action invokes. Specify either apexClass or fileBasedApexClass but not both. Applies only if the type is `Apex`.
`inputDataProcessor`	**Field Type** string **Description** The optional Integration Procedure that processes the sent data. Applies only if the type is `LowCode`.

Field Name	Description
`integrationProviderAttributes`	**Field Type** IntegrationProviderAttr[] **Description** Custom attributes that store data associated with an Integration Definition.
`outputDataProcessor`	**Field Type** string **Description** The optional Integration Procedure that processes the returned data. Applies only if the type is `LowCode`.
`providerLabel`	**Field Type** string **Description** Required. A meaningful name for the Integration Definition.
`type`	**Field Type** DefinitionType (enumeration of type string) **Description** Required. What the Integration Definition calls, either an Apex class or an external service. Values are: • `Apex` • `LowCode`

IntegrationProviderAttr

A custom attribute that stores data associated with an Integration Definition.

Field Name	Description
`dataType`	**Field Type** AttrDataType (enumeration of type string) **Description** Required. The data type of the attribute. Values are: • `Date`

Field Name	Description
	• `DateTime` • `Double` • `Integer` • `Percentage` • `String` • `Boolean`
`dateTimeValue`	**Field Type** dateTime **Description** The value of the attribute if the `dataType` is `DateTime`.
`dateValue`	**Field Type** date **Description** The value of the attribute if the `dataType` is `Date`.
`description`	**Field Type** string **Description** A meaningful explanation of the attribute.
`developerName`	**Field Type** string **Description** Required. A system name for the attribute.
`doubleValue`	**Field Type** double **Description** The value of the attribute if the `dataType` is `Double`.
`integerValue`	**Field Type** int **Description** The value of the attribute if the `dataType` is `Integer`.
`label`	**Field Type** string

Field Name	Description
	Description Required. A meaningful name for the attribute.
`percentageValue`	**Field Type** double **Description** The value of the attribute if the `dataType` is `Percentage`.
`required`	**Field Type** boolean **Description** Required. Specifies whether the attribute is required.
`stringValue`	**Field Type** string **Description** The value of the attribute if the `dataType` is `String`.
`trueOrFalseValue`	**Field Type** boolean **Description** The value of the attribute if the `dataType` is `Boolean`.

Declarative Metadata Sample Definition

The following is an example of an IntegrationProviderDef component.

```
<?xml version="1.0" encoding="UTF-8"?>
<IntegrationProviderDef xmlns="http://soap.sforce.com/2006/04/metadata">
   <developerName>EmailUpdate</developerName>
   <providerLabel>EmailUpdate</providerLabel>
   <type>Apex</type>
   <apexClass>SendEmailUpdate</apexClass>
   <integrationProviderAttributes>
      <developerName>EmailAddress</developerName>
      <label>EmailAddress</label>
      <dataType>String</dataType>
      <stringValue>person@example.com</stringValue>
      <required>true</required>
   </integrationProviderAttributes>
</IntegrationProviderDef>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>IntegrationProviderDef</name>
    </types>
    <version>57.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

IPAddressRange

Represents a range of IP addresses to include in or exclude from the specified feature.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

IP Address Range components have the suffix `.IPAddressRange` and are stored in the `IPAddressRanges` folder.

Version

IPAddressRange components are available in API version 52.0 and later.

Special Access Rules

To access IpAddressRange, enable the HtmlEmail permission in your org.

Fields

Field Name	Description
`Description`	**Field Type** string **Description** Not required. The description of the IP address range. For example, the name of the company that owns the IP address range.

Field Name	Description
`developerName`	**Field Type** string **Description** Not required. Gives you a way to distinguish ipAddressRange entries among developers in your org.
`endIpAddress`	**Field Type** string **Description** The end of the IP address range. Must be an IPv4 or IPv6 Internet address and equal to or greater than the `startIpAddress`.
`ipAddressFeature`	**Field Type** picklist **Description** The feature that uses the range of IP addresses. Possible values are: • `EmailIpFiltering` (default) —Filter email engagement activities such as email opens and email clicks.
`ipAddressUsageScope`	**Field Type** picklist **Description** Whether the specified IP addresses are included or excluded. Possible values are: • `Exclusion` • `Inclusion`
`isProtected`	**Field Type** boolean **Description** Whether the specified IP address range is protected. The default is `false`.
`masterLabel`	**Field Type** string **Description** Master label for the IP address range. This internal label doesn't get translated.
`startIpAddress`	**Field Type** string **Description** The start of the IP address range. Must be an IPv4 or IPv6 Internet address and equal to or smaller than the `endIpAddress`.

Declarative Metadata Sample Definition

The following is an example of an `ipAddressName` component.

```
<?xml version="1.0" encoding="UTF-8"?>
<IPAddressRange xmlns="http://soap.sforce.com/2006/04/metadata">
    <description>Filter emails from google.com</description>
    <endIpAddress>221.224.222.158</endIpAddress>
    <ipAddressFeature>EmailIpFiltering</ipAddressFeature>
    <ipAddressUsageScope>Exclusion</ipAddressUsageScope>
    <masterLabel>MasterLabelValue</masterLabel>
    <startIpAddress>221.224.0.158</startIpAddress>
    <isProtected>false</isProtected>
</IPAddressRange>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>IPAddressRange</name>
    </types>
    <version>1.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type doesn't support the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

KeywordList

Represents a list of keywords used in Experience Cloud site moderation. This keyword list is a type of moderation criteria that defines offensive language or inappropriate content that you don't want in your site.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

This type extends the Metadata metadata type and inherits its `fullName` field.

Keep the following things in mind when creating keyword list criteria:

- Your org can have up to 30 keyword list criteria. This limit is per org, not per Experience Cloud site.
- A keyword list can have up to 2,000 keywords.
- Capitalization and trailing punctuation are ignored when matching your keywords to user-generated content. For example, if your criteria includes *`BadWord`*, it's matched when a user types *`BADWORD`* or *`badword.`*

File Suffix and Directory Location

KeywordList components have the suffix `.keywords` and are stored in the `moderation` directory of the corresponding package directory. The file name format follows *`site_name`*`.`*`keyword_list_developer_name`*`.keywords`.

Version

KeywordList components are available in API version 36.0 and later.

Special Access Rules

To view, create, edit, and delete a keyword list, you need the Manage Experiences or Create and Set Up Experiences permission. As of Spring '20 and later, only users with permission to edit moderation rules can access this object.

Fields

Field Name	Field Type	Description
`Description`	string	A description of the keyword list.
`keywords`	Keyword[]	The keywords you want moderate in your Experience Cloud site.
`masterLabel`	string	Required. Label for the keyword list.

Keyword

Keywords in the keyword list.

Field Name	Field Type	Description
`keyword`	string	Required. Keywords you want to moderate. • Keywords can only be up to 100 characters and can include letters, numbers, spaces, and special characters. • Wildcard characters aren't supported.

Declarative Metadata Sample Definition

The following is an example of a KeywordList component.

```
<?xml version="1.0" encoding="UTF-8"?>
<KeywordList xmlns="http://soap.sforce.com/2006/04/metadata">
   <masterLabel>Bad Word List</masterLabel>
   <description>List of bad words updated by Joe in Nov 2015.</description>
   <keywords>
     <keyword>bad-word</keyword>
   </keywords>
   <keywords>
     <keyword>b a d w o r d</keyword>
   </keywords>
   <keywords>
     <keyword>b@dword</keyword>
   </keywords>
</KeywordList>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
   <types>
     <name>KeywordList</name>
     <members>site1.badword_list</members>
   </types>
   <version>36.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

Layout

Represents the metadata associated with a page layout. For more information, see Page Layouts in Salesforce Help.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

This type extends the Metadata metadata type and inherits its `fullName` field.

Note: To edit the Ideas layout, specify it by name in the `package.xml` file. In `package.xml`, use this code to retrieve the Ideas layout.

```
<types>
    <members>Idea-Idea Layout</members>
    <name>Layout</name>
</types>
```

File Suffix and Directory Location

Layouts are stored in the `layouts` directory of the corresponding package directory. The extension is `.layout`.

Note: Retrieving a component of this metadata type in a project makes the component appear in any Profile and PermissionSet components that are retrieved in the same package.

Version

Layouts are available in API version 13.0 and later.

Fields

This metadata type represents the valid values that define a page layout.

Field Name	Field Type	Description
`customButtons`	string[]	The custom buttons for this layout. Each button is a reference to a WebLink on the same object. For example, a ButtonLink refers to a Weblink on the same standard or custom object named ButtonLink.
`customConsoleComponents`	CustomConsoleComponents	Represents custom console components (Visualforce pages, lookup fields, or related lists; Canvas apps not available) on a page layout. Custom console components only display in the Salesforce console.
`emailDefault`	boolean	Only relevant if showEmailCheckbox is set; indicates the default value of that checkbox.
`excludeButtons`	string[]	List of standard buttons to exclude from this layout. For example, `<excludeButtons>Delete</excludeButtons>` excludes the **Delete** button from this layout.
`feedLayout`	FeedLayout	Represents the values that define the feed view of a feed-based page layout. Feed-based layouts are available on Account, Case, Contact, Lead, Opportunity, custom, and external objects. They include a feed view and a detail view.
`headers`	LayoutHeader[] (enumeration of type string)	Layout headers are currently only used for tagging, and only appear in the UI if tagging is enabled. Valid string values are: • `PersonalTagging`—tag is private to user. • `PublicTagging`—tag is viewable any other user who can access the record.
`layoutSections`	LayoutSection[]	The main sections of the layout containing fields, s-controls, and custom links. The order here determines the layout order.
`miniLayout`	MiniLayout	A mini layout is used in the mini view of a record in the Console tab, hover details, and event overlays.
`multilineLayoutFields`	string[]	Fields for the special multiline layout fields that appear in OpportunityProduct layouts. These fields are otherwise similar to `miniLayoutFields`.
`platformActionList`	PlatformActionList	The list of actions and their order that appear in the Salesforce mobile app action bar for the layout. This field is available in API version 34.0 and later.
`quickActionList`	QuickActionList	The list of quick actions that display in the full Salesforce site for the page layout. This field is available in API version 28.0 and later.

Field Name	Field Type	Description
relatedContent	RelatedContent	The Related Content section of the page layout. This field is available in API version 29.0 and later.
relatedLists	RelatedListItem[]	The related lists for the layout, listed in the order they appear in the user interface.
relatedObjects	string[]	The list of related objects that appears in the mini view of the console. In database terms, these objects are foreign key fields on the object for the layout. For more information, see Choose Related Objects for the Agent Console's Mini View in Salesforce Help.
runAssignmentRulesDefault	boolean	Only relevant if showRunAssignmentRulesCheckbox is set; indicates the default value of that checkbox.
showEmailCheckbox	boolean	Only allowed on Case, CaseClose, and Task layouts. If set, a checkbox appears to show email.
showHighlightsPanel	boolean	If set, the highlights panel displays on pages in the Salesforce console. This field is available in API version 22.0 and later.
showInteractionLogPanel	boolean	If set, the interaction log displays on pages in the Salesforce console. This field is available in API version 22.0 and later.
showKnowledgeComponent	boolean	Only allowed on Case layouts. If set, the Knowledge sidebar displays on cases in the Salesforce console. This field is available in API version 20.0 and later.
showRunAssignmentRulesCheckbox	boolean	Only allowed on Case, Lead, and Account objects. If set, a checkbox appears on the page to show assignment rules.
showSolutionSection	boolean	Only allowed on CaseClose layout. If set, the built-in solution information section shows up on the page.
showSubmitAndAttachButton	boolean	Only allowed on Case layout. If set, the **Submit & Add Attachment** button displays on case edit pages to portal users in the Customer Portal.
summaryLayout	SummaryLayout	Controls the appearance of the highlights panel in Salesforce Classic, which summarizes key fields in a grid at the top of a page layout, when Case Feed is enabled. This field is available in API version 18.0 and later.

CustomConsoleComponents

Represents custom console components (Visualforce pages, lookup fields, or related lists; Canvas apps not available) on a page layout. Custom console components only appear in the Salesforce console. Available in API version 25.0 and later.

Field Name	Field Type	Description
primaryTabComponents	PrimaryTabComponents	Represents custom console components on primary tabs in the Salesforce console. Available in API version 25.0 and later.
subtabComponents	SubtabComponents	Represents custom console components on subtabs in the Salesforce console. Available in API version 25.0 and later.

PrimaryTabComponents

Represents custom console components on primary tabs in the Salesforce console. Available in API version 25.0 and later.

Field Name	Field Type	Description
component	ConsoleComponent[]	Represents a custom console component (Visualforce page, lookup field, or related lists; Canvas apps not available) on a section of a page layout. Custom console components only appear in the Salesforce console. This field is available in API version 29.0 and earlier.
containers	Container[]	Represents a location and style to display more than one custom console component on the sidebars of the Salesforce console. You can specify up to five components for each of the four locations (left, right, top, and bottom). This field is available in API version 30.0 and later.

ConsoleComponent

Represents a custom console component (Visualforce page, lookup field, or related lists; Canvas apps not available) on a section of a page layout. Custom console components only appear in the Salesforce console. Available in API version 25.0 and later.

Field Name	Field Type	Description
height	int	Required for components with a location of top or bottom. The height of the custom console component. The value must be specified in pixels and be greater than 0 but less than 999.
location	string	Required. The location of the custom console component on the page layout. Valid values are right, left, top, and bottom. A component can have one location for each page layout.
visualforcePage	string	Required. The unique name of the custom console component. For example, ConsoleComponentPage.
width	int	Required for components with a location of left or right. The width of the custom console component. The value must be specified in pixels and be greater than 0 but less than 999.

Container

Represents a location and style to display more than one custom console component in the sidebars of the Salesforce console. For example, you can show multiple components in the right sidebar of the console with a style of either stack, tabs, or accordion. Available in API version 30.0 and later.

Field Name	Field Type	Description
`height`	int	Required for components with a location of top or bottom. The height of the components' container. The `unit` field determines the unit of measurement, in pixels or percent.
`isContainerAutoSizeEnabled`	boolean	Required. If set to `true`, stacked console components in the sidebars autosize vertically. Set to `true` by default for newly created console components. Available in API version 32.0 and later.
`region`	string	Required. The location of the components' container. Valid values include: • `right` • `left` • `top` • `bottom`
`sidebarComponents`	SidebarComponent[]	Represents a specific custom console component to display in the components' container.
`style`	string	Required. The style of the container to display multiple components. Valid values include: • `stack`—a content area with multiple frames. • `tabs`—a single content area with a list of multiple panels. • `accordian`—a collapsible content area.
`unit`	string	Required. The unit of measurement, in pixels or percent, for the height or width of the components' container. Pixel values are simply the number of pixels, for example, `500`, and must be greater than 0 but less than 999. Percentage values must include the percent sign, for example, `20%`, and must be greater than 0 but less than 100.
`width`	int	Required for components with a location of right or left. The width of the components' container. The `unit` field determines the unit of measurement, in pixels or percent.

SidebarComponent

Represents a specific custom console component to display in a container that hosts multiple components in one of the sidebars of the Salesforce console. You can specify up to five components for each of the four container locations (left, right, top, and bottom). Available in API version 30.0 and later.

Field Name	Field Type	Description
`componentType`	string	Specifies the component type. Valid values are `KnowledgeOne`, `Lookup`, `Milestones`, `RelatedList`, `Topics`, `Files`, and `CaseExperts`. This field is available in API version 31.0 and later. The `Files` and `CaseExperts` values are available in API version 32.0 and later. Case Experts is available through a pilot program.
`createAction`	string	If the component is a lookup field, the name of the quick action used to create a record. This field is available in API version 42.0 and later.
`enableLinking`	boolean	If the component is a lookup field, lets users associate a record with this field. This field is available in API version 42.0 and later. If false, the createAction and updateAction can't be retrieved.
`height`	int	Required for components with a location of top or bottom. The height of the component in the container. The `unit` field determines the unit of measurement, in pixels or percent.
`label`	string	The name of the component as it appears to console users. Available for components in a container with the style of tabs or accordion.
`lookup`	string	If the component is a lookup field, the name of the field.
`page`	string	If the component is a Visualforce page, the name of the Visualforce page.
`relatedlists`	RelatedList[]	If the component is a related list, the name of the list. This field is available in API version 31.0 and later.
`unit`	string	The unit of measurement, in pixels or percent, for the height or width of the component in the container. Pixel values are simply the number of pixels, for example, `500`, and must be greater than 0 but less than 999. Percentage values must include the percent sign, for example, `20%`, and must be greater than 0 but less than 100.
`updateAction`	string	If the component is a lookup field, the name of the quick action used to update a record. This field is available in API version 42.0 and later.
`width`	int	Required for components with a location of right or left. The width of the component in the container. The `unit` field determines the unit of measurement, in pixels or percent.

RelatedList

Represents related list custom components on the sidebars of the Salesforce console. Available in API version 31.0 and later.

Field Name	Field Type	Description
`hideOnDetail`	boolean	If set to `true`, the related list is hidden from detail pages where it appears as a component to prevent duplicate information from showing.
`name`	string	The name of the component as it appears to console users.

SubtabComponents

Represents custom console components on subtabs in the Salesforce console. Available in API version 25.0 and later.

Field Name	Field Type	Description
`component`	ConsoleComponent[]	Represents a custom console component (Visualforce page, lookup field, or related lists; Canvas apps not available) on a section of a page layout. Custom console components only appear in the Salesforce console. This field is available in API version 29.0 and earlier.
`containers`	Container[]	Represents a location and style to display more than one custom console component on the sidebars of the Salesforce console. You can specify up to five components for each of the four locations (left, right, top, and bottom). This field is available in API version 30.0 and later.

FeedLayout

Represents the values that define the feed view of a feed-based page layout. Feed-based layouts are available on Account, Case, Contact, Lead, Opportunity, custom, and external objects. They include a feed view and a detail view. Available in API version 30.0 and later.

Field Name	Field Type	Description
`autocollapsePublisher`	boolean	Specifies whether the publisher is automatically collapsed when the page loads (`true`) or not (`false`).
`compactFeed`	boolean	Specifies whether the feed-based page layout uses a compact feed (`true`) or not (`false`). If set to `true`, feed items on the page are collapsed by default, and the feed view has an updated design.
`feedFilterPosition`	FeedLayoutFilterPosition (enumeration of type string)	Where the feed filters list is included in the layout. Valid values are: • `centerDropDown`—as a dropdown list in the center column. • `leftFixed`—as a fixed list in the left column. • `leftFloat`—as a floating list in the left column.
`feedFilters`	FeedLayoutFilter[]	The individual filters displayed in the feed filters list.
`fullWidthFeed`	boolean	Specifies whether the feed expands horizontally to take up all available space on the page (`true`) or not (`false`).
`hideSidebar`	boolean	Specifies whether the sidebar is hidden (`true`) or not (`false`).
`leftComponents`	FeedLayoutComponent[]	The individual components displayed in the left column of the feed view.

Field Name	Field Type	Description
`rightComponents`	FeedLayoutComponent[]	The individual components displayed in the right column of the feed view.

FeedLayoutComponent

Represents a component in the feed view of a feed-based page layout. Available in API version 30.0 and later.

Field Name	Field Type	Description
`componentType`	FeedLayoutComponentType (enumeration of type string)	Required. The type of component. Valid values are: • `HelpAndToolLinks`—icons that link to the help topic for the page, the page layout, and, the printable view of the page. Available only on Case layouts. • `CustomButtons`—a custom button. • `Following`—an icon that toggles between a Follow button (if the user viewing a record doesn't already follow it) and a Following indicator (if the user viewing a record does follow it). • `Followers`—a list of users who follow the record. • `CustomLinks`—a custom link. • `Milestones`—the milestone tracker, which lets users see the status of a milestone on a case. Available only on Case layouts. • `Topics`—a list of topics related to the record. • `CaseUnifiedFiles`—a list of all files that are attached to the case. • `Visualforce`—a custom Visualforce component.
`height`	int	The height, in pixels, of the component. Doesn't apply to `standardComponents`
`page`	string	The name of a Visualforce page being used as a custom component.

FeedLayoutFilter

Represents a feed filter option in the feed view of a feed-based page layout. A filter must have only `standardFilter` or `feedItemType` set. Available in API version 30.0 and later.

Field Name	Field Type	Description
`feedFilterName`	string	The name of a CustomFeedFilter component. Names are prefixed with the name of the parent object. For example, *`Case.MyCustomFeedFilter`*.
`feedFilterType`	FeedLayoutFilterType (enumeration of type string)	The type of filter. Valid values are: • `AllUpdates`—shows all feed items on a record.

Field Name	Field Type	Description
		• `FeedItemType`—shows feed items only for a particular type of activity on the record.
`feedItemType`	FeedItemType (enumeration of type string)	The type of feed item to display. Valid values are: • `ActivityEvent`—feed items related to activity on tasks and events associated with a case. Available only on Case layouts. • `AdvancedTextPost`–feed items related to group announcements posted on a feed. This value is available in API version 31.0 and later. • `AnnouncementPost`–Not used. • `ApprovalPost`—feed items related to approvals that are submitted on a feed. • `AttachArticleEvent`—feed items for activity related to attaching articles to cases. Available only on Case layouts. • `BasicTemplateFeedItem`—Not used. • `CallLogPost`—feed items for activity from the Log a Call action. Available only on layouts for objects that support Activities (tasks and events). • `CanvasPost`—feed items related to posts that a canvas app makes on a feed. • `CaseCommentPost`—feed items for activity from the Case Note action. Available only on Case layouts. • `ChangeStatusPost`—feed items for activity from the Change Status action. Available only on Case layouts. • `ChatTranscriptPost`—feed items for activity related to attaching Chat transcripts to cases. Available only on Case layouts. • `CollaborationGroupCreated`—feed items related to creating a public group. • `CollaborationGroupUnarchived`—Not used. • `ContentPost`—feed items related to attaching a file to a post. • `CreatedRecordEvent`—feed items related to creating a record from the publisher. • `DashboardComponentSnapshot`—feed items related to posting a dashboard snapshot on a feed. • `EmailMessageEvent`—feed items for activity from the Email action. Available only on Case layouts. • `FacebookPost`—Not used. • `LinkPost`—feed items related to attaching a URL to a post. • `MilestoneEvent`—feed items for changes to the milestone status on a case. Available only on Case layouts. • `PollPost`—feed items related to posting a poll on a feed.

Field Name	Field Type	Description
		• `ProfileSkillPost`—feed items related to skills added to a user's Chatter profile. This value is available in API version 31.0 and later. • `QuestionPost`—feed items related to posting a question on a feed. This value is available in API version 31.0 and later. • `ReplyPost`—feed items for activity from the Portal action. Available only on Case layouts. • `RypplePost`—feed items related to creating a Thanks badge in WDC. • `SocialPost`—feed items for activity on Twitter from the Social Post action. • `TextPost`—feed items for creating a text post from the publisher. • `TrackedChange`—feed items related to a change or group of changes to a tracked field. • `UserStatus`—Not used.

MiniLayout

Represents a mini view of a record in the Console tab, hover details, and event overlays.

Field Name	Field Type	Description
`fields`	string[]	The fields for the mini-layout, listed in the order they appear in the UI. Fields that appear here must appear in the main layout.
`relatedLists`	RelatedListItem[]	The mini related list, listed in the order they appear in the UI. You can't set sorting on mini related lists. Fields that appear here must appear in the main layout.

LayoutSection

LayoutSection represents a section of a page layout, such as the Custom Links section.

Field Name	Field Type	Description
`customLabel`	boolean	Indicates if this section's label is custom or standard (built-in). Custom labels can be any text, but must be translated. Standard labels have a predefined set of valid values, for example System Information, which are automatically translated.
`detailHeading`	boolean	Controls if this section appears in the detail page. In the UI, this setting corresponds to the checkbox in the section details dialog.
`editHeading`	boolean	Controls if this section appears in the edit page.

Field Name	Field Type	Description
`label`	string	The label; either standard or custom, based on the `customLabel` flag.
`layoutColumns`	LayoutColumn[]	The columns of the layout, depending on the style. 1, 2, or 3 columns, ordered left to right, are possible.
`style`	LayoutSectionStyle (enumeration of type string)	The style of the layout: • `TwoColumnsTopToBottom` - Two columns, tab goes top to bottom • `TwoColumnsLeftToRight` - Two columns, tab goes left to right • `OneColumn` - One column • `CustomLinks` - Contains custom links only

LayoutColumn

LayoutColumn represents the items in a column within a layout section.

Field Name	Field Type	Description
`layoutItems`	LayoutItem[]	The individual items within a column (ordered from top to bottom).
`reserved`	string	This field is reserved for Salesforce. The field resolves an issue with some SOAP libraries. Any value entered in the field is ignored.

LayoutItem

LayoutItem represents the valid values that define a layout item. An item must have only one of the following values set: component, customLink, field, s-control, page, analyticsCloudComponent, or reportChartComponent.

Field Name	Field Type	Description
`behavior`	UiBehavior (enumeration of type string)	Determines the field behavior. Valid string values: • `Edit`—The layout field can be edited but isn't required. • `Required`—The layout field can be edited and is required. • `Readonly`—The layout field is read-only. Explicitly specifying UiBehavior for Knowledge articles results in an exception.
`canvas`	string	Reference to a canvas app. This field is available in API version 31.0 and later.

Field Name	Field Type	Description
`component`	string	Reference to a component. Value must be *`sfa:socialCard`*. This field is available in API version 30.0 and later. This field is allowed only inside a `RelatedContentItem`. *`sfa:socialCard`* is supported only on page layouts for contacts, accounts, and leads.
`customLink`	string	The `customLink` reference. This field is allowed only inside a `CustomLink layoutSection`.
`emptySpace`	boolean	Controls if this layout item is a blank space.
`field`	string	The field name reference, relative to the layout object, for example `Description` or `MyField__c`.
`height`	int	For s-control and pages only, the height in pixels.
`page`	string	Reference to a Visualforce page.
`analyticsCloudComponent`	AnalyticsCloudComponentLayoutItem	Refers to a CRM Analytics dashboard that you can add to a standard or custom object page. This field is available in API version 34.0 and later.
`reportChartComponent`	ReportChartComponentLayoutItem	Refers to a report chart that you can add to a standard or custom object page.
`scontrol`	string	Reference to an s-control.
`showLabel`	boolean	For s-control and pages only, whether to show the label.
`showScrollbars`	boolean	For s-control and pages only, whether to show scrollbars.
`width`	string	For s-control and pages only, the width in pixels or percent. Pixel values are simply the number of pixels, for example, `500`. Percentage values must include the percent sign, for example, `20%`.

AnalyticsCloudComponentLayoutItem

Represents the settings for a CRM Analytics dashboard on a standard or custom page. Available in API version 34.0 and later.

Field Name	Field Type	Description
`assetType`	string	Required. Specifies the type of CRM Analytics asset to add. The available asset type is `dashboard`.
`devName`	string	Required. Unique development name of the dashboard to add.
`error`	string	Error string; only populated if an error occurred in the underlying dashboard.

Field Name	Field Type	Description
`filter`	string	Communicates initial dashboard filters for mapping data fields in the dashboard to the object's fields, so that the dashboard shows only the data that's relevant for the record being viewed.
`height`	int	Specifies the height of the dashboard, in pixels. The default is `400`.
`hideOnError`	boolean	Controls whether users see a dashboard that has an error. When this attribute is set to `true`, if the dashboard has an error, the dashboard doesn't appear on the page. When set to `false`, the dashboard appears but doesn't show any data except the error. An error can happen when a user doesn't have access to CRM Analytics or to the dashboard. The default is `true`.
`showSharing`	boolean	If set to `true`, and the dashboard is shareable, then the dashboard shows the Share icon. Users can click the icon to open the Share dialog and post or download from the dashboard. If set to `false`, the dashboard doesn't show the Share icon. This field is available in API version 37.0 and later.
`showTitle`	boolean	If `true`, includes the dashboard's title above the dashboard. If `false`, the dashboard appears without a title. The default is `true`.
`width`	string	Specifies the width of the dashboard, in pixels or percent. Pixel values are simply the number of pixels, for example, `500`. Percentage values must include the percent sign, for example, `20%`. The default is `100%`.

ReportChartComponentLayoutItem

Represents the settings for a report chart on a standard or custom page.

Field Name	Field Type	Description
`cacheData`	boolean	Indicates whether to use cached data when displaying the chart. When the attribute is set to `true`, data is cached for 24 hours. If the attribute is set to `false`, the report isn't run every time the page is refreshed. This field is available in API version 29.0 and later.
`contextFilterableField`	string	Unique development name of the field by which a report chart is filtered to return data relevant to the page. If set, the ID field for the parent object of the page or report type is the chart data filter. The parent object for the report type and the page must match for a chart to return relevant data.
`error`	string	Error string; only populated if an error occurred in the underlying report. This field is available in API version 31.0 and later.
`hideOnError`	boolean	Controls whether users see a chart that has an error. When there's an error and this attribute is set, the chart doesn't show any data except the error. An error can happen for many reasons, such as when a user doesn't

Field Name	Field Type	Description
		have access to fields used by the chart or a chart has been removed from the report. Set the attribute to `true` to hide the chart from a page on error. This field is available in API version 29.0 and later.
`includeContext`	boolean	If `true`, filters the report chart to return data that's relevant to the page.
`reportName`	string	Unique development name of a report that includes a chart.
`showTitle`	boolean	If `true`, applies the title from the report to the chart.
`size`	ReportChartComponentSize (enumeration of type string)	The chart size is medium when no value is specified. Valid values: • `SMALL` • `MEDIUM` • `LARGE`

PlatformActionList

PlatformActionList represents the list of actions and their order that appear in the Salesforce mobile app action bar for the layout. Available in API version 34.0 and later.

Field Name	Field Type	Description
`actionListContext`	PlatformActionListContext (enumeration of type string)	Required. The context of the action list. Valid values are: • `Assistant` • `BannerPhoto` • `Chatter` • `Dockable` • `FeedElement` • `Flexipage` • `Global` • `ListView` • `ListViewDefinition` • `ListViewRecord` • `Lookup` • `MruList` • `MruRow` • `ObjectHomeChart` • `Photo` • `Record` • `RecordEdit` • `RelatedList`

Field Name	Field Type	Description
		• `RelatedListRecord`
`platformActionListItems`	PlatformActionListItem[]	The actions in the PlatformActionList.
`relatedSourceEntity`	string	When the `ActionListContext` is RelatedList or RelatedListRecord, this field represents the API name of the related list to which the action belongs.

PlatformActionListItem

PlatformActionListItem represents an action in the PlatformActionList. Available in API version 34.0 and later.

Field Name	Field Type	Description
`actionName`	string	The API name for the action in the list.
`actionType`	PlatformActionType (enumeration of type string)	The type of action. Valid values are: • `ActionLink`—An indicator on a feed element that targets an API, a web page, or a file, represented by a button in the Salesforce Chatter feed UI. • `CustomButton`—When clicked, opens a URL or a Visualforce page in a window or executes JavaScript. • `InvocableAction` • `ProductivityAction`—Productivity actions are predefined and attached to a limited set of objects. Productivity actions include Send Email, Call, Map, View Website, and Read News. Except for the Call action, you can't edit productivity actions. • `QuickAction`—A global or object-specific action. • `StandardButton`—A predefined Salesforce button such as New, Edit, and Delete.
`sortOrder`	int	The placement of the action in the list.
`subtype`	string	The subtype of the action. For quick actions, the subtype is `QuickActionType`. For custom buttons, the subtype is `WebLinkTypeEnum`. For action links, subtypes are `Api`, `ApiAsync`, `Download`, and `Ui`. Standard buttons and productivity actions have no subtype.

QuickActionList

QuickActionList represents the list of actions associated with the page layout. Available in API version 28.0 and later.

Field Name	Field Type	Description
`quickActionListItems`	QuickActionListItem[]	Array of zero or more QuickActionList objects.

QuickActionListItem

QuickActionListItem represents an action in the QuickActionList. Available in API version 28.0 and later.

Field Name	Field Type	Description
`quickActionName`	string	The API name of the action.

RelatedContent

RelatedContent represents the Mobile Cards section of the page layout. Available in API version 29.0 and later.

Field Name	Field Type	Description
`relatedContentItems`	RelatedContentItem[]	A list of layout items in the Mobile Cards section of the page layout.

RelatedContentItem

RelatedContentItem represents an individual item in the RelatedContentItem list. Available in API version 29.0 and later.

Field Name	Field Type	Description
`layoutItem`	LayoutItem	An individual LayoutItem in the Mobile Cards section.

RelatedListItem

RelatedListItem represents a related list in a page layout.

Field Name	Field Type	Description
`customButtons`	string[]	A list of custom buttons that are used on the related list.
`excludeButtons`	string[]	A list of buttons that are excluded from the related list.
`fields`	string[]	A list of fields that are displayed in the related list. Retrieval of standard fields on related lists uses aliases instead of field or API names. For example, the `Fax`, `Mobile`, and `Home Phone` fields are retrieved as Phone2, Phone3, and Phone4, respectively.
`quickActions`	string[]	A list of quick actions that are used on the related list.
`relatedList`	string	Required. The name of the related list.
`sortField`	string	The name of the field that is used for sorting.
`sortOrder`	SortOrder (enumeration of type string)	If the `sortField` is set, the `sortOrder` field determines the sort order. • `Asc` - Sort in ascending order • `Desc` - Sort in descending order

SummaryLayout

When Case Feed is enabled, controls the appearance of the highlights panel in Salesforce Classic, which summarizes key fields in a grid at the top of a page layout. Available in API version 25.0 and later.

Field Name	Field Type	Description
`masterLabel`	string	Required. The name of the layout label.
`sizeX`	int	Required. Number of columns in the highlights pane, from 1 through 4 (inclusive).
`sizeY`	int	Required. Number of rows in each column, either 1 or 2.
`sizeZ`	int	Reserved for future use. If provided, the setting is visible to users.
`summaryLayoutItems`	SummaryLayoutItem[]	Controls the appearance of an individual field and its column and row position within the highlights panel grid, when Case Feed is enabled. At least one is required.
`summaryLayoutStyle`	SummaryLayoutStyle (enumeration of type string)	Highlights panel style. Valid string values are: • Default • QuoteTemplate • DefaultQuoteTemplate • CaseInteraction • QuickActionLayoutLeftRight (Available in API version 28.0 and later.) • QuickActionLayoutTopDown (Available in API version 28.0 and later.)

SummaryLayoutItem

Controls the appearance of an individual field and its column and row position within the highlights panel grid, when Case Feed is enabled. You can have two fields per each grid in a highlights panel. Available in API version 25.0 and later.

Field Name	Field Type	Description
`customLink`	string	The `customLink` reference, if the item is a custom link.
`field`	string	The field name reference, relative to the page layout. Must be a standard or custom field that also exists on the detail page.
`posX`	int	Required. The item's column position in the highlights panel grid. Must be within the range of `sizeX`.
`posY`	int	Required. The item's row position in the highlights panel grid. Must be within the range of `sizeY`.
`posZ`	int	Reserved for future use. If provided, the setting is visible to users.

Declarative Metadata Sample Definition

This sample defines a page layout.

```
<?xml version="1.0" encoding="UTF-8"?>
<Layout xmlns="http://soap.sforce.com/2006/04/metadata">
   <customConsoleComponents>
    <primaryTabComponents>
       <container>
           <region>left</region>
          <style>Stack</style>
          <unit>Pixel</unit>
          <width>101</width>
          <sidebarComponent>
                <width>60</width>
                <page>simplepage1</page>
                <unit>Percentage</unit>
          </sidebarComponent>
          <sidebarComponent>
                <width>40</width>
                <page>Hello_World</page>
                <unit>Percentage</unit>
          </sidebarComponent>
       </container>
    </primaryTabComponents>
    <subtabComponents>
       <component>
          <location>top</location>
          <visualforcePage>ConsoleComponentPage2</visualforcePage>
          <height>200</height>
       </component>
    </subtabComponents>
   </customConsoleComponents>
    <customButtons>ButtonLink</customButtons>
    <layoutSections>
        <editHeading>true</editHeading>
        <label>Information</label>
        <layoutColumns>
            <layoutItems>
                <behavior>Required</behavior>
                <field>Name</field>
            </layoutItems>
            <layoutItems>
                <height>180</height>
                <scontrol>LayoutSControl</scontrol>
                <showLabel>true</showLabel>
                <showScrollbars>true</showScrollbars>
                <width>50%</width>
            </layoutItems>
            <layoutItems>
                <reportChartComponent>
                   <contextFilterableField>CUST_ID</contextFilterableField>
                   <includeContext>true</includeContext>
                   <reportName>Open_Accounts_by_Cases</reportName>
                   <showTitle>false</showTitle>
```

```
                <size>LARGE</size>
            <reportChartComponent>
        </layoutItems>
    </layoutColumns>
    <layoutColumns>
        <layoutItems>
            <behavior>Edit</behavior>
            <field>OwnerId</field>
        </layoutItems>
        <layoutItems>
            <behavior>Edit</behavior>
            <field>CurrencyIsoCode</field>
        </layoutItems>
    </layoutColumns>
    <style>TwoColumnsTopToBottom</style>
</layoutSections>
<layoutSections>
    <editHeading>true</editHeading>
    <label>System Information</label>
    <layoutColumns>
        <layoutItems>
            <behavior>Readonly</behavior>
            <field>CreatedById</field>
        </layoutItems>
        <layoutItems>
            <behavior>Readonly</behavior>
            <field>Alpha1__c</field>
        </layoutItems>
        <layoutItems>
            <height>200</height>
            <page>mcanvasPage</page>
            <showLabel>true</showLabel>
            <showScrollbars>false</showScrollbars>
            <width>100%</width>
        </layoutItems>
    </layoutColumns>
    <layoutColumns>
        <layoutItems>
            <behavior>Readonly</behavior>
            <field>LastModifiedById</field>
        </layoutItems>
        <layoutItems>
            <behavior>Edit</behavior>
            <field>TextArea__c</field>
        </layoutItems>
    </layoutColumns>
    <style>TwoColumnsTopToBottom</style>
</layoutSections>
<layoutSections>
    <customLabel>true</customLabel>
    <detailHeading>true</detailHeading>
    <label>Custom Links</label>
    <layoutColumns>
        <layoutItems>
```

```
                <customLink>CustomWebLink</customLink>
            </layoutItems>
        </layoutColumns>
        <style>CustomLinks</style>
    </layoutSections>
     <quickActionList>
        <quickActionListItems>
            <quickActionName>FeedItem.TextPost</quickActionName>
        </quickActionListItems>
        <quickActionListItems>
            <quickActionName>FeedItem.ContentPost</quickActionName>
        </quickActionListItems>
        <quickActionListItems>
            <quickActionName>FeedItem.LinkPost</quickActionName>
        </quickActionListItems>
        <quickActionListItems>
            <quickActionName>FeedItem.PollPost</quickActionName>
        </quickActionListItems>
    </quickActionList>
    <relatedContent>
        <relatedContentItems>
            <layoutItem>
                <component>sfa:socialPanel</component>
            </layoutItem>
        <relatedContentItems>
    </relatedContent>
    <miniLayoutFields>Name</miniLayoutFields>
    <miniLayoutFields>OwnerId</miniLayoutFields>
    <miniLayoutFields>CurrencyIsoCode</miniLayoutFields>
    <miniLayoutFields>Alpha1__c</miniLayoutFields>
    <miniLayoutFields>TextArea__c</miniLayoutFields>
    <miniRelatedLists>
        <relatedList>RelatedNoteList</relatedList>
    </miniRelatedLists>
    <relatedLists>
        <fields>StepStatus</fields>
        <fields>CreatedDate</fields>
        <fields>OriginalActor</fields>
        <fields>Actor</fields>
        <fields>Comments</fields>
        <fields>Actor.Alias</fields>
        <fields>OriginalActor.Alias</fields>
        <relatedList>RelatedProcessHistoryList</relatedList>
    </relatedLists>
    <relatedLists>
        <relatedList>RelatedNoteList</relatedList>
    </relatedLists>
</Layout>
```

This example shows a layout using `<summaryLayout>`.

```
<?xml version="1.0" encoding="UTF-8"?>
<Layout xmlns="http://soap.sforce.com/2006/04/metadata">
    <layoutSections>
        <editHeading>true</editHeading>
```

```
        <label>System Information</label>
        <layoutColumns>
            <layoutItems>
                <behavior>Readonly</behavior>
                <field>CreatedById</field>
            </layoutItems>
            <layoutItems>
                <behavior>Required</behavior>
                <field>Name</field>
            </layoutItems>
        </layoutColumns>
        <layoutColumns>
            <layoutItems>
                <behavior>Readonly</behavior>
                <field>LastModifiedById</field>
            </layoutItems>
        </layoutColumns>
        <style>TwoColumnsTopToBottom</style>
    </layoutSections>
    <summaryLayout>
        <masterLabel>Great Name</masterLabel>
        <sizeX>4</sizeX>
        <sizeY>2</sizeY>
        <summaryLayoutItems>
            <posX>0</posX>
            <posY>0</posY>
            <field>Name</field>
        </summaryLayoutItems>
    </summaryLayout>
</Layout>
```

This example shows a feed-based layout.

```
<Layout>
...
    <feedLayout>
        <leftComponents>
            <componentType>customLinks</componentType>
        </leftComponents>
        <rightComponents>
            <componentType>follow</componentType>
        </rightComponents>
        <rightComponents>
            <componentType>followers</componentType>
        </rightComponents>
        <rightComponents>
            <componentType>visualforce</componentType>
            <page>accountCustomWidget</page>
            <height>200</height>
        </rightComponents>
        <hideSidebar>true</hideSidebar>
        <feedFilterPosition>centerDropDown</feedFilterPosition>
        <feedFilters>
    <feedFilerType>allUpdates</feedFilerType>
        </feedFilters>
```

```
        <feedFilters>
    <feedFilerType>feedItemType</feedFilerType>
    <feedItemType>CallLogPost</feedItemType>
        </feedFilters>
        <feedFilters>
    <feedFilerType>feedItemType</feedFilerType>
    <feedItemType>TextPost</feedItemType>
        </feedFilters>
    </feedLayout>
...
</Layout>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

LearningItemType

Represents a custom exercise type that an Enablement user takes in an Enablement program in the Guidance Center. A custom exercise type also requires a corresponding LearningItem record for the Guidance Center and corresponding EnblProgramTaskDefinition and EnblProgramTaskSubCategory records for when admins create a program in Program Builder.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

LearningItemType components have the suffix `.learningItemType` and are stored in the `learningItemTypes` folder.

Version

LearningItemType components are available in API version 62.0 and later.

Special Access Rules

- For Enablement admins to create, update, and delete Enablement programs, the Design and Deliver Enablement Programs permission is required. This permission is enabled by default as part of the Manage Enablement Essentials permission set, which comes with the Enablement add-on license.
- For users who take Enablement programs, the Take Enablement Programs permission is required. This permission is enabled by default as part of the Use Enablement Programs permission set, which comes with the Enablement add-on license.

Important: Custom exercises aren't compatible with Partner Enablement programs.

Fields

Field Name	Description
`apexEvaluationHandler`	**Field Type** string **Description** The ID of the Apex class that specifies how progress and completion of the custom exercise is assessed when users take the program in the Guidance Center.
`apexSerializerDeserializer`	**Field Type** string **Description** The ID of the Apex class that specifies how data related to the custom exercise type is retrieved and deployed with change sets or managed packages.
`customField`	**Field Type** string **Description** Required. The programmatic name of a custom lookup field on the LearningItem object that references the custom object used with this custom exercise. For example, if a custom exercise type shows a screen flow, maybe the custom object's name is `ScreenFlow_Object__c` and the custom field on LearningItem is named `ScreenFlow_Field__c`. For details, see Implement Custom Exercise Types for Enablement Programs in the *Sales Programs and Partner Tracks with Enablement Developer Guide*. This field is unique within your organization.
`customObject`	**Field Type** string **Description** Required. The programmatic name of the custom object used with this custom exercise. For example, if a custom exercise type shows a screen flow, maybe the custom object's name is `ScreenFlow_Field__c`. For details, see Implement Custom Exercise Types for Enablement Programs in the *Sales Programs and Partner Tracks with Enablement Developer Guide*. This field is unique within your organization.
`developerName`	**Field Type** string **Description** Required. The unique programmatic name for the LearningItemType record.

Field Name	Description
`icon`	**Field Type** string **Description** Required. The icon to use for the custom exercise type in the Guidance Center. Use the format ***`iconType:iconName`***, where the values correspond to icon categories and names from the Salesforce Lightning Design System. • ***`iconType`*** is the type of icon, such as `standard` or `doctype`. • ***`iconName`*** is the icon name, such as `flow` or `slide`. For example, to use the Standard type Flow icon, this value is `standard:flow`.
`lightningComponentDefinition`	**Field Type** string **Description** Required. The ID of the Lightning Web Component used to show the custom exercise's content when a user opens the exercise in the Guidance Center. This field sets the value of the `LightningComponentName` field on the LearningItemType object.
`masterLabel`	**Field Type** string **Description** Required. A user-friendly name for the LearningItemType, which is defined when it's created.

Declarative Metadata Sample Definition

The following is an example of a LearningItemType component for a custom exercise type that shows a screen flow.

```
<?xml version="1.0" encoding="UTF-8"?>
<LearningItemType xmlns="http://soap.sforce.com/2006/04/metadata">
    <apexEvaluationHandler>ScreenFlowEvaluationHandler</apexEvaluationHandler>
   <apexSerializerDeserializer>ScreenFlowSerializerDeserializer</apexSerializerDeserializer>

    <customField>ScreenFlow_Field__c</customField>
    <customObject>ScreenFlow_Object__c</customObject>
    <developerName>ScreenFlowLearningItemType</developerName>
    <icon>standard:flow</icon>
    <lightningComponentDefinition>screenFlowViewer</lightningComponentDef>
    <masterLabel>Screen Flow Exercise</masterLabel>
</LearningItemType>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>ScreenFlowLearningItemType</members>
        <name>LearningItemType</name>
    </types>
    <version>62.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

Letterhead

Represents formatting options for the letterhead in an email template. A letterhead defines the logo, page color, and text settings for your HTML email templates. Use letterheads to ensure a consistent look and feel in your company's emails.

For more information, see "Create Classic Letterheads for Email Templates" in the Salesforce online help. This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

The file suffix for letterheads is `.letter` and components are stored in the `letterhead` directory of the corresponding package directory.

Version

Letterheads are available in API version 12.0 and later.

Fields

With the exception of logo, and horizontal and vertical alignment, all of these fields are required.

Field Name	Field Type	Description
`available`	boolean	Required. Indicates whether this letterhead can be used (`true`) or not (`false`), for example, in an email template.
`backgroundColor`	string	Required. The background color, in hexadecimal, for example `#FF6600`.
`bodyColor`	string	Required. The body color in hexadecimal.

Field Name	Field Type	Description
`bottomLine`	LetterheadLine (enumeration of type string)	Required. The style for the bottom line. Valid style values include: • `color`. The color of the line in hexadecimal, as a string value. • `height`. The height of the line, as an int value.
`description`	string	Text description of how this letterhead differs from other letterheads.
`fullName`	string	The internal name of the letterhead, based on the `name`, but with white spaces and special characters escaped out for validity.
`footer`	LetterheadHeaderFooter	Required. The style for the footer.
`header`	LetterheadHeaderFooter	Required. The style for the header.
`middleLine`	LetterheadLine	Required. The style for the middle border line in your letterhead. Valid style values include: • `color`. The color of the line in hexadecimal, as a string value. • `height`. The height of the line, as an int value.
`name`	string	Required. The name of the letterhead.
`topLine`	LetterheadLine	Required. The style for the top horizontal line below the header. Valid style values include: • `color`. The color of the line in hexadecimal, as a string value. • `height`. The height of the line, as an int value.

LetterheadHeaderFooter

LetterheadHeaderFooter represents the properties of a header or footer.

Field	Field Type	Description
`backgroundColor`	string	Required. The background color of the header or footer in hexadecimal format.
`height`	DashboardComponent[]	Required. The height of the header or footer.
`horizontalAlignment`	LetterheadHorizontalAlignment (enumeration of type string)	The horizontal alignment of the header or footer. Valid values are: • `None` • `Left` • `Center`

Field	Field Type	Description
		• Right
logo	string	The logo which is a reference to a document, for example MyFolder/MyDocument.gif.
verticalAlignment	LetterheadVerticalAlignment (enumeration of type string)	The vertical alignment of the header or footer. Valid values are: • None • Top • Middle • Bottom

LetterheadLine

LetterheadLine represents the properties of a line.

Field	Field Type	Description
color	string	Required. The color of the line in hexadecimal format.
height	int	Required. The height of the line.

Declarative Metadata Sample Definition

```
<?xml version="1.0" encoding="UTF-8"?>
<Letterhead xmlns="http://soap.sforce.com/2006/04/metadata">
    <available>true</available>
    <backgroundColor>#CCCCCC</backgroundColor>
    <bodyColor>#33FF33</bodyColor>
    <bottomLine>
        <color>#3333FF</color>
        <height>5</height>
    </bottomLine>
    <description>INITIAL</description>
    <footer>
        <backgroundColor>#FFFFFF</backgroundColor>
        <height>100</height>
        <horizontalAlignment>Left</horizontalAlignment>
        <verticalAlignment>Top</verticalAlignment>
    </footer>
    <header>
        <backgroundColor>#FFFFFF</backgroundColor>
        <height>100</height>
        <horizontalAlignment>Left</horizontalAlignment>
        <verticalAlignment>Top</verticalAlignment>
    </header>
    <middleLine>
        <color>#AAAAFF</color>
```

```
        <height>5</height>
    </middleLine>
    <name>SimpleLetterheadLabel</name>
    <topLine>
        <color>#FF99FF</color>
        <height>5</height>
    </topLine>
</Letterhead>
```

Wildcard Support in the Manifest File

This metadata type doesn't support the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

LightningBolt

Represents the definition of a Lightning Bolt Solution, which can include custom apps, flow categories, and Experience Builder templates. This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

LightningBolt components have the suffix `.lightningBolt` and are stored in the `lightningBolts` folder.

Version

LightningBolt components are available in API version 43.0 and later.

Special Access Rules

To add Experience Builder templates to a Lightning Bolt Solution, enable digital experiences in your org.

Fields

Field Name	Field Type	Description
`category`	LightningBoltCategory (enumeration of type string)	Required. The primary industry that the Lightning Bolt Solution is aimed at. Valid values are: • `Communications` • `Education` • `FinancialServices` • `GeneralBusiness` • `Government`

Field Name	Field Type	Description
		• `HealthcareLifeSciences` • `HighTech` • `Manufacturing` • `Media` • `Nonprofits` • `ProfessionalServices` • `RealEstate` • `Retail` • `TravelTransportationHospitality`
`lightningBoltFeatures`	LightningBoltFeatures[]	The list of feature descriptions of this Lightning Bolt Solution.
`lightningBoltImages`	LightningBoltImages[]	The list of images of this Lightning Bolt Solution.
`lightningBoltItems`	LightningBoltItems[]	The list of items (custom apps, flow categories, and Experience Builder templates) that comprise this Lightning Bolt Solution.
`masterLabel`	string	Required. The label of the Lightning Bolt Solution, which appears on the solution detail page.
`publisher`	string	Required. The name of the partner org associated with this Lightning Bolt Solution.
`summary`	string	Required. The summary description of the Lightning Bolt Solution.

LightningBoltFeatures

Represents the list of feature descriptions of a Lightning Bolt Solution.

Field Name	Field Type	Description
`description`	string	A description of the feature of the Lightning Bolt Solution.
`order`	int	Required. An integer specifying the position of this feature relative to others in the list. 1 is the first position, and 4 is the max position.
`title`	string	Required. The title of the feature, which appears on the solution detail page.

LightningBoltImages

Represents the list of images of a Lightning Bolt Solution.

Field Name	Field Type	Description
`image`	string	Required. The developer name of the `ContentAsset` type, which is used as a preview image for this Lightning Bolt Solution.

Field Name	Field Type	Description
order	int	Required. An integer specifying the position of this image relative to others in the list. 1 is the first position, and 3 is the max position.

LightningBoltItems

Represents the list of items (custom apps, flow categories, and Experience Builder templates) that comprise a Lightning Bolt Solution.

Field Name	Field Type	Description
name	string	Required. The name of the item, which appears on the solution detail page.
type	string	Required. The type of the item included in the Lightning Bolt Solution. Valid values are: • CommunityTemplateDefinition • CustomApplication • FlowCategory

Declarative Metadata Sample Definition

The following is an example of a LightningBolt component.

```
<?xml version="1.0" encoding="UTF-8"?>
<LightningBolt xmlns="http://soap.sforce.com/2006/04/metadata"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <category>Sales</category>
    <lightningBoltFeatures>
        <description>bb</description>
        <order>1</order>
        <title>aa</title>
    </lightningBoltFeatures>
    <lightningBoltImages>
        <image>prm1</image>
        <order>1</order>
    </lightningBoltImages>
    <lightningBoltItems>
        <name>PolaConsole</name>
        <type>CustomApplication</type>
    </lightningBoltItems>
    <lightningBoltItems>
        <name>Banking_Service_Console</name>
        <type>CustomApplication</type>
    </lightningBoltItems>
    <lightningBoltItems>
        <name>Banking_Service_Portal</name>
        <type>CommunityTemplateDefinition</type>
    </lightningBoltItems>
    <lightningBoltItems>
        <name>Banking_Sales_Portal</name>
```

```
        <type>CommunityTemplateDefinition</type>
    </lightningBoltItems>
    <lightningBoltItems>
        <name>myorgdev__updatebenefits</name>
        <type>FlowCategory</type>
    </lightningBoltItems>
    <masterLabel>BoltTe</masterLabel>
    <publisher>aaaa</publisher>
    <summary>This is a summary.</summary>

</LightningBolt>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>BoltTe</members>
        <name>LightningBolt</name>
    </types>
    <version>43.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

LightningComponentBundle

Represents a Lightning web component bundle. A bundle contains Lightning web component resources.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Special Access Rules

LightningComponentBundle components can be created only in orgs with defined namespaces.

As of Summer '20 and later, only your Salesforce org's internal users can access this type.

For more information on packaging a Lightning web component bundle, see the Second-Generation Managed Packaging Developer Guide.

Fields

Field Name	Field Type	Description
`apiVersion`	double	A double value that binds the component to a Salesforce API version.

Field Name	Field Type	Description
`capabilities`	Capabilities[]	A list of capabilities. A capability is something that a component can do, as opposed to a target, which defines where you can use a component. Available in API version 48.0 and later.
`description`	string	A description of the Lightning web component.
`isExplicitImport`	boolean	Indicates whether imports between files are done explicitly by the developer (`true`) or implicitly by the framework (`false`).
`isExposed`	boolean	If `true`, the component is available to other namespaces. If `true` and a `targets` value is also provided, the component is available to Salesforce builders such as Lightning App Builder and Experience Builder. If `false`, the component isn't available to builders and other namespaces.
`lwcResources`	LwcResources[]	A list of resources inside a bundle.
`masterLabel`	string	The component title that appears in the list view.

`targetConfigs`	base64Binary	Configurations for each target. Each target is a Lightning page type. For example, this configuration allows a Lightning web component to be used on a Contact record page in Lightning App Builder.
		`<targetConfigs>`
		`  <targetConfig targets="lightning__RecordPage">`
		`  <objects>`
		`  <object>Contact</object>`
		`  </objects>`
		`  </targetConfig>`
		`</targetConfigs>`

`targets`	Targets[]	A list of targets where the Lightning web component can be used, such as in Lightning App Builder or Experience Builder sites.

Capabilities

Represents a list of capabilities. A capability is something that a component can do, as opposed to a target, which defines where you can use a component. Available in API version 48.0 and later. For more information, see XML Configuration File Elements.

Field	Field Type	Description
`capability`	string	Specifies something that a component can do. Valid values are: • `lightningCommunity__RelaxedCSP` • `lightning__dynamicComponent` • `lightning__ServerRenderable` • `lightning__ServerRenderableWithHydration` • `lightning__ServiceCloudVoiceToolkitApi`

LwcResources

Represents a list of resources inside a LightningComponentBundle.

Field	Field Type	Description
`lwcResource`	LwcResource	A resource inside a LightningComponentBundle.

LwcResource

Represents a resource inside a LightningComponentBundle.

Field	Field Type	Description
`filePath`	string	Required. The file path of a resource.
`source`	base64Binary	Required. The content of a resource.

Targets

Represents a list of supported containers for a Lightning web component. For more information, see XML Configuration File Elements.

Field	Field Type	Description
`target`	string	Specifies the type of Lightning page the component can be added to in the builders, such as in Lightning App Builder, Experience Builder, Flow Builder, or Document Builder. For valid values, see XML Configuration File Elements: target.

Declarative Metadata Sample Definition

This `package.xml` file retrieves all the LightningComponentBundle components in an org.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>LightningComponentBundle</name>
    </types>
    <version>45.0</version>
</Package>
```

In the retrieved `zip` file, each Lightning web component is nested under an `lwc` folder.

This example shows the directory structure in the zip file of one component with a name of `hello`.

```
lwc
    hello
        hello.html
```

```
        hello.js
        hello.js-meta.xml
```

Here are the contents of the files in the `hello` directory.

Content of `hello.html`:

```
<template>

    <lightning-card title="Hello" icon-name="custom:custom14">
        <div class="slds-m-around_medium">
            Hello, {greeting}!
        </div>
    </lightning-card>

</template>
```

Content of `hello.js`:

```
import { LightningElement } from 'lwc';

export default class Hello extends LightningElement {
    greeting = 'World';
}
```

Content of `hello.js-meta.xml`.

```
<?xml version="1.0" encoding="UTF-8"?>
<LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">
    <apiVersion>45.0</apiVersion>
    <isExposed>true</isExposed>
    <targets>
        <target>lightning__AppPage</target>
        <target>lightning__RecordPage</target>
        <target>lightning__HomePage</target>
    </targets>
</LightningComponentBundle>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

LightningExperienceTheme

Represents the details of a custom theme, including the BrandingSet. Themes enable admins to specify configurable attributes, such as three colors and five images. The colors and some of the images override SLDS token values and influence the generation of `app.css`.

To activate a custom theme with Metadata API, set the `activeThemeField` on the LightningExperienceSettings component to the API name of the LightningExperienceTheme.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

LightningExperienceTheme components have the suffix `.lightningExperienceTheme` and are stored in the `lightningExperienceThemes` folder.

Version

LightningExperienceTheme components are available in API version 42.0 and later.

Special Access Rules

The LightningExperieceTheme type is available when the S1DesktopAllowed permission is enabled in your org.

Fields

Field Name	Field Type	Description
`defaultBrandingSet`	string	Required. The ID of the BrandingSet properties associated with this LightningExperienceTheme.
`description`	string	The optional description text of this LightningExperienceTheme. Limited to 1000 characters.
`designSystemVersion`	LightningDesignSystemVersion (enumeration of type string)	Represents the version of Salesforce Lightning Design System (SLDS) on which the theme is built. Valid values are: • `SLDS_v1` • `SLDS_v2` If you don't define a value, the default value is `SLDS_v1`. Available in API version 64.0 and later.
`masterLabel`	string	Required. The label for this LightningExperienceTheme, which displays in Setup. Limited to 70 characters.
`shouldOverrideLoadingImage`	boolean	If `true`, the LightningExperienceTheme overrides the splash screen image.

Declarative Metadata Sample Definition

The following is an example of a LightningExperienceTheme component. See BrandingSet on page 466 for an example of the BrandingSet component.

```
<?xml version="1.0" encoding="UTF-8"?>
<LightningExperienceTheme xmlns="http://soap.sforce.com/2006/04/metadata">
    <defaultBrandingSet>SummerCelebrationBrand</defaultBrandingSet>
    <description>Theme for summer celebration week.</description>
    <masterLabel>Summer Celebration</masterLabel>
```

```
    <shouldOverrideLoadingImage>false</shouldOverrideLoadingImage>
</LightningExperienceTheme>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>LEXTHEMINGThemeName</members>
        <name>BrandingSet</name>
    </types>
    <types>
     <members>Summer Celebration</members>
     <name>LightningExperienceTheme</name>
    </types>
    <version>42.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

LightningMessageChannel

Represents the metadata associated with a Lightning Message Channel. A Lightning Message Channel represents a secure channel to communicate across UI technologies, such as Lightning Web Components, Aura Components, and Visualforce.

This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Note: Before you include a Lightning Message Channel in a managed package, review these considerations.

- To pass the AppExchange Security Review, you must set the `isExposed` field to `false`.
- If you set the `isExposed` field to `true`, you can't change the value to `false` at a later time. This consideration applies to Lightning Message Channels in managed packages and Lightning Message Channels that other components reference.
- Visualforce supports only Lightning Message Channels where `isExposed` is `true`, so managed packages with a Lightning Message Channel in Visualforce can't pass the AppExchange Security Review. See Considerations and Limitations in the *Visualforce Developer Guide*.

File Suffix and Directory Location

LightningMessageChannel components have the suffix `.messageChannel` and are stored in the `messageChannels` folder.

Version

LightningMessageChannel components are available in API version 47.0 and later.

Fields

Field Name	Field Type	Description
`description`	string	The description of the Lightning Message Channel.
`isExposed`	boolean	Indicates whether a Lightning Message Channel is exposed to components in other namespaces (`true`) or not (`false`). The default value is `false`.
`lightningMessageFields`	LightningMessageField on page 1372[]	A list of message payload fields for a given Lightning Message Channel.
`masterLabel`	string	Required. The label for a Lightning Message Channel.

LightningMessageField

Represents a message payload field for a given Lightning Message Channel.

Field Name	Field Type	Description
`description`	string	The description for a Lightning Message Field.
`fieldName`	string	Required. Unique identifier of the Lightning Message Field.

Declarative Metadata Sample Definition

Here's a simple example of a LightningMessageChannel component.

```
<?xml version="1.0" encoding="UTF-8"?>
<LightningMessageChannel xmlns="http://soap.sforce.com/2006/04/metadata">
    <masterLabel>SampleMessageChannel</masterLabel>
    <isExposed>true</isExposed>
    <description>This is a sample Lightning Message Channel.</description>
</LightningMessageChannel>
```

Here's an example of a LightningMessageChannel component with LightningMessageFields.

```
<?xml version="1.0" encoding="UTF-8"?>
<LightningMessageChannel xmlns="http://soap.sforce.com/2006/04/metadata">
    <masterLabel>SampleMessageChannel</masterLabel>
    <isExposed>true</isExposed>
    <description>This is a sample Lightning Message Channel.</description>
    <lightningMessageFields>
        <fieldName>recordId</fieldName>
        <description>This is the record Id that changed</description>
    </lightningMessageFields>
    <lightningMessageFields>
        <fieldName>recordData</fieldName>
        <description>The current data representing the record that changed</description>
    </lightningMessageFields>
</LightningMessageChannel>
```

Here's an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>LightningMessageChannel</name>
    </types>
    <version>47.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SEE ALSO:

Lightning Web Components Developer Guide: Communicate Across the DOM with Lightning Message Service

Second-Generation Managed Packaging Developer Guide: Components Available in Managed Packages

LightningOnboardingConfig

Represents the feedback provided when users switch from Lightning Experience to Salesforce Classic. Admins can customize the question, how frequently the form appears, and where the feedback is stored in Chatter from the Adoption Assistance page in Lightning Experience Setup. This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

LightningOnboardingConfig components have the suffix `.lightningOnboardingConfig` and are stored in the `LightningOnboardingConfigs` folder.

Version

LightningOnboardingConfig components are available in API version 49.0 and later.

Special Access Rules

See Switch to Salesforce Classic Feedback Form in Salesforce Help for details.

Fields

Field Name	Field Type	Description
`collaborationGroup`	string	Required. The ID of the Chatter Group where the user feedback is posted.

Field Name	Field Type	Description
customQuestion	string	Text of the custom question added by the admin. Maximum of 1,000 characters.
feedbackFormDaysFrequency	int	Required. The number of days between showing the feedback form when a user switches between Lightning Experience and Salesforce Classic. A value of 0 indicates that the form is shown for every switch. Maximum of 30.
isCustom	boolean	Required. Indicates if a feedback form includes a custom question (true) or not (false).
masterLabel	string	Required. The label of the in-app guidance. Maximum of 80 characters.
promptDelayTime	int	Required. Indicates the amount of time, in seconds, to delay between instances of all in-app content, both custom content created by org and standard content created by Salesforce. Minimum of 0 hours and 0 minutes. Maximum of 99 hours and 59 minutes.
sendFeedbackToSalesforce	boolean	Required. Indicates if the user feedback can be shared with Salesforce (true) or not (false). Even if the feedback isn't shared with Salesforce, the feedback is shared in the Chatter Group chosen when customizing the feedback form. The default is false.

Declarative Metadata Sample Definition

The following is an example of a LightningOnboardingConfig component.

```
<?xml version="1.0" encoding="UTF-8"?>
<LightningOnboardingConfig xmlns="http://soap.sforce.com/2006/04/metadata">
    <collaborationGroup>{Org ID}</collaborationGroup>
    <customQuestion>Please take a minute to tell us why you're switching.</customQuestion>

    <feedbackFormDaysFrequency>0</feedbackFormDaysFrequency>
    <isCustom>true</isCustom>
    <masterLabel>Feedback Form</masterLabel>
    <promptDelayTime>3600</promptDelayTime>
    <sendFeedbackToSalesforce>true</sendFeedbackToSalesforce>
</LightningOnboardingConfig>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character * (asterisk) in the package.xml manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

LightningTypeBundle

Represents a custom Lightning type. Use this type to override the default user interface to create a customized appearance of responses on the custom agent's action input and output. Deploy this bundle to your organization to implement the overrides.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Structure and Directory Location

LightningTypeBundle components are stored in the `lightningTypes` folder.

Here's an example of the `LightningTypeBundle` structure.

```
+--myMetadataPackage
    +--lightningTypes (1)
        +--TypeName (2)
           +--schema.json (3)
           +--lightningDesktopGenAi (4)
              +--editor.json (5) OR +--renderer.json (6)
```

The bundle includes the following resources:

- The `lightningTypes` folder (1) contains a folder for each created custom Lightning type in the format `TypeName` (2).
- Each custom lightning type folder contains a `schema.json` file (3) that defines the JSON schema that drives the custom Lightning type validation.
- If applicable, the custom lightning type folder also contains a `lightningDesktopGenAi` folder (4) with two files that indicate the optional artifacts needed for the `lightningDesktopGenAi` channel. Configure these files to override the default UI of a custom Lightning type when it's used in an agent action:
  - The `editor.json` file (5) has custom user interface and editor information.
  - The `renderer.json` file (6) has custom user interface and renderer information.

Version

LightningTypeBundle components are available in API version 64.0 and later.

Fields

Field Name	Description
`description`	**Field Type** string **Description** Describes the lightning type.
`masterLabel`	**Field Type** string

Field Name	Description
	Description Required. Represents the name of a LightningTypeBundle which is defined when the LightningTypeBundle is created.
`resources`	**Field Type** LightningTypeBundleResource[] **Description** The list of resource files in the `lightningTypes` folder.

LightningTypeBundleResource

Represents a resource inside a LightningTypeBundle.

Field Name	Description
`fileName`	**Field Type** string **Description** Required. Name of the resource file.
`filePath`	**Field Type** string **Description** Required. Path of the resource file.
`source`	**Field Type** base64Binary **Description** Required. The JSON content of the resource.

Declarative Metadata Sample Definition

This `package.xml` file retrieves all the LightningTypeBundle components in an org.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>LightningTypeBundle</name>
    </types>
    <version>64.0</version>
</Package>
```

In the retrieved `.zip` file, each custom Lightning type is nested under a `lightningTypes` folder.

This example shows the directory structure in the `.zip` file of a custom Lightning type named `flightResponse`:

```
+--lightningTypes
    +--flightResponse
        +--schema.json
        +--lightningDesktopGenAi
            +--renderer.json
```

In this example, the custom Lightning type `flightResponse` is a complex type that references an Apex class named `Flight`. The Apex class `Flight` is nested under the parent class `FlightAgent`.

```
public class FlightAgent {
    public class Flight {
        public String flightId;
        public Integer numLayovers;
        public Boolean isPetAllowed;
        public Long price;
        public Double discountPercentage;
    }
    public class Filter {
        public Long price;
        public Double discountPercentage;
    }
}
```

Here are the contents of the files in the `flightResponse` directory.

The following sample code shows the contents of the `schema.json` file.

```
{
  "title": "My Flight Response",
  "description": "My Flight Response",
  "lightning:type": "@apexClassType/c__FlightAgent$Flight"
}
```

The `lightningDesktopGenAi` folder (optional) includes a `renderer.json` file that overrides the default UI of the custom Lightning type `flightResponse` when you use the `Flight` Apex class as an output parameter for an agent action.

Contents of the `renderer.json` file.

```
{
  "renderer": {
    "componentOverrides": {
      "$": {
        "definition": "c/flightDetails"
      }
    }
  }
}
```

Note: flightDetails is a custom LWC component referenced in `renderer.json` file.

This example shows the directory structure in the `.zip` file of a custom Lightning type named `flightFilter`:

```
+--lightningTypes
    +--flightFilter
```

```
        +--schema.json
        +--lightningDesktopGenAi
            +--editor.json
```

In this example, the custom Lightning type `flightFilter` is a complex type that references an Apex class named `Filter`. The Apex class `Filter` is nested under the parent class `FlightAgent`.

```
public class FlightAgent {
    public class Flight {
        public String flightId;
        public Integer numLayovers;
        public Boolean isPetAllowed;
        public Long price;
        public Double discountPercentage;
    }
    public class Filter {
        public Long price;
        public Double discountPercentage;
    }
}
```

Here are the contents of the files in the `flightFilter` directory.

The following sample code shows the contents of the `schema.json` file.

```
{
  "title": "Flight Filter",
  "description": "Flight Filter",
  "lightning:type": "@apexClassType/c__FlightAgent$Filter"
}
```

The `lightningDesktopGenAi` folder (optional) includes an `editor.json` file that overrides the default UI of the custom Lightning type `flightFilter` when you use the `Flight` Apex class as an input parameter for an agent action.

Contents of the `editor.json` file.

```
{
  "editor": {
    "componentOverrides": {
      "$": {
        "definition": "c/flightFilter"
      }
    }
  }
}
```

Note: flightFilter is a custom LWC component referenced in `editor.json` file.

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SEE ALSO:

https://developer.salesforce.com/docs/einstein/genai/guide/lightning-types.html

LiveChatAgentConfig

Represents the configuration of an organization's Chat deployment, such as how many chats can be assigned to an agent and whether chat sounds are enabled.

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

LiveChatAgentConfig configurations are referenced in the `<developer_name>.liveChatAgentConfig` file in the `liveChatAgentConfigs` directory.

Version

LiveChatAgentConfig is available in API version 28.0 and later.

Fields

Field Name	Field Type	Description
`assignments`	AgentConfigAssignments	Specifies how agent configurations are assigned to Chat users. Agent configurations can be assigned to sets of users or sets of profiles.
`autoGreeting`	string	Specifies the greeting that displays when a customer begins a chat with an agent.
`capacity`	int	Specifies the maximum number of chats in which an agent can be engaged at a time.
`criticalWaitTime`	int	Specifies the number of seconds an agent can wait to answer an engaged chat before the chat tab flashes to alert the agent to answer it.
`customAgentName`	string	Specifies the custom name for an agent, if one has been set. Available in API version 29.0 and later.
`disableTransferConferenceGreeting`	boolean	Indicates whether the greeting is disabled for agents during chat transfer and chat conferencing (`true`) or not (`false`). Available in API version 53.0 and later.

Field Name	Field Type	Description
`enableAgentFileTransfer`	boolean	Indicates whether file transfer is enabled for agents (`true`) or not (`false`). Available in API version 31.0 and later.
`enableAgentSneakPeek`	boolean	Specifies whether a supervisor can see the content of an agent's message before they send it to a customer (`true`) or not (`false`).
`enableAssistanceFlag`	boolean	Indicates whether agents can raise an assistance flag to notify a supervisor that they need help. Available in API version 35.0 and later.
`enableAutoAwayOnDecline`	boolean	Indicates whether an agent appears as "away" (`true`) or not (`false`) when an agent declines a chat with a customer.
`enableAutoAwayOnPushTimeout`	boolean	Indicates whether an agent appears as "away" (`true`) or not (`false`) when a chat request that's been pushed to the agent times out. Available in API version 34.0 and later.
`enableChatConferencing`	boolean	Indicates whether chat conferencing is enabled for agents (`true`) or not (`false`). Available in API version 34.0 and later.
`enableChatMonitoring`	boolean	Indicates whether chat monitoring is enabled for support supervisors (`true`) or not (`false`). Available in API version 29.0 and later.
`enableChatTransferToAgent`	boolean	Indicates whether agents can transfer a chat to another agent (`true`) or not (`false`). Available in API version 36.0 and later.
`enableChatTransferToButton`	boolean	Indicates whether agents can transfer a chat to a button (`true`) or not (`false`). Available in API version 36.0 and later.
`enableChatTransferToSkill`	boolean	Indicates whether agents can transfer a chat to a skill group (`true`) or not (`false`). Available in API version 36.0 and later.
`enableLogoutSound`	boolean	Indicates whether a sound plays (`true`) or not (`false`) when an agent logs out of Chat.
`enableNotifications`	boolean	Indicates whether notifications of incoming chats appear for agents (`true`) or not (`false`).
`enableRequestSound`	boolean	Indicates whether a sound plays (`true`) or not (`false`) when a customer requests to chat with an agent.
`enableSneakPeek`	boolean	Indicates whether previews of customers' messages are displayed as customers type (`true`) or not (`false`) in the agent's Chat window. Available in API version 29.0 and later.

Field Name	Field Type	Description
`enableVisitorBlocking`	boolean	Indicates whether an agent can block a visitor by IP address (`true`) or not (`false`). Available in API version 34.0 and later.
`enableWhisperMessage`	boolean	Indicates whether support supervisors can send whisper messages to agents during a chat (`true`) or not (`false`). Available in API version 29.0 and later.
`label`	string	Required. Specifies the name of the configuration for agents' default chat settings.
`supervisorDefaultAgentStatusFilter`	SupervisorAgentStatusFilter (enumeration of type string)	Specifies the Chat status for filtering the Agent Status list in the Supervisor Panel. Valid values are: • `Online` • `Away` • `Offline` Available in API version 29.0 and later.
`supervisorDefaultButtonFilter`	string	Specifies the default button for filtering the Agent Status list in the Supervisor Panel. Available in API version 29.0 and later.
`supervisorDefaultSkillFilter`	string	Specifies the default skill for filtering the Agent Status list in the Supervisor Panel. Available in API version 29.0 and later.
`supervisorSkills`	SupervisorAgentConfigSkills	Specifies the list of agent skills that are assigned to a supervisor, as specified in their assigned Chat configuration. Available in API version 29.0 and later.
`transferableButtons`	AgentConfigButtons	Specifies the list of chat buttons that agents can transfer chats to. Available in API version 31.0 and later.
`transferableSkills`	AgentConfigSkills	Specifies the list of skill groups that agents can transfer chats to. Available in API version 31.0 and later.

AgentConfigAssignments

Represents the assignments of an organization's profiles and users to a Chat configuration.

Field Name	Field Type	Description
`profiles`	AgentConfigProfileAssignments	Specifies the profiles that are associated with a specific agent configuration.
`users`	AgentConfigUserAssignments	Specifies the users that are associated with a specific agent configuration.

AgentConfigButtons

Represents the chat buttons that agents who are associated with the Chat configuration can transfer chats to.

Field Name	Field Type	Description
`button`	string[]	Specifies the chat buttons that agents can transfer chats to.

AgentConfigProfileAssignments

Represents the profiles associated with a specific Chat configuration.

Field Name	Field Type	Description
`profile`	string	Specifies the custom name of the profile associated with a specific agent configuration.

AgentConfigSkills

Represents the skill groups that agents who are associated with the Chat configuration can transfer chats to.

Field Name	Field Type	Description
`skill`	string[]	Specifies the skill groups that agents can transfer chats to.

AgentConfigUserAssignments

Represents the users associated with a specific Chat configuration.

Field Name	Field Type	Description
`user`	string	Specifies the username of the user associated with a specific agent configuration.

SupervisorAgentConfigSkills

Represents the agent skills associated with a supervisor's Chat configuration. Available in API version 29.0 and later.

Field Name	Field Type	Description
`skill`	string	Specifies the agent skills available for filtering the Agent Status list in the Supervisor Panel.

Declarative Metadata Sample Definition

This is a sample of a `liveChatAgentConfig` file.

```
<?xml version="1.0" encoding="UTF-8"?>
<LiveChatAgentConfig xmlns="http://soap.sforce.com/2006/04/metadata">
    <label>My Agent Configuration 1</label>
    <autoGreeting>Hi, how can I help you?</autoGreeting>
    <capacity>5</capacity>
    <enableAutoAwayOnDecline>true</enableAutoAwayOnDecline>
    <enableLogoutSound>true</enableLogoutSound>
    <enableNotifications>true</enableNotifications>
    <enableRequestSound>true</enableRequestSound>
    <enableSneakPeek>true</enableSneakPeek>
    <enableWhisperMessage>true</enableWhisperMessage>
    <assignments>
        <profiles>
            <profile>standard</profile>
        </profiles>
        <users>
            <user>jdoe@acme.com</user>
        </users>
    </assignments>
</LiveChatAgentConfig>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

LiveChatButton

Represents a Chat deployment's settings for the button that customers click to chat with an agent and the chat window, such as the label that appears on the button and the pre-chat form that appears before a chat begins. This type extends the Metadata metadata type and inherits its `fullName` field.

Chats routed with Omni-Channel aren't supported in the Metadata API.

File Suffix and Directory Location

LiveChatButton on page 1383 configurations are stored in the `<developer_name>.liveChatButton` file in the `liveChatButtons` directory.

Version

LiveChatButton on page 1383 is available in API version 28.0 and later.

Fields

Field Name	Field Type	Description
`animation`	LiveChatButtonPresentation (enumeration of type string)	The type of animation for a chat invitation. Valid values are: • `Slide` • `Fade` • `Appear` • `Custom`
`autoGreeting`	string	The customized greeting message that the customer receives when an agent accepts a chat request from the chat button or invitation. Available in API version 29.0 and later.
`chasitorIdleTimeout`	int	Specifies the amount of idle time before the chat times out. The idle time starts being counted after the agent sends the last chat message. Available in API version 35.0 and later.
`chasitorIdleTimeoutWarning`	int	Specifies the amount of idle time before a warning appears. The idle time starts being counted after the agent sends the last chat message. Available in API version 35.0 and later.
`chatPage`	string	Specifies the page that hosts your chat if that page differs from the Chat window.
`customAgentName`	string	The agent's name as it appears to customers in the chat window. Available in API version 29.0 and later.
`deployments`	LiveChatButtonDeployments	Specifies the deployments associated with the button.
`enableQueue`	boolean	Indicates whether queuing is enabled (`true`) or not (`false`).
`inviteEndPosition`	LiveChatButtonInviteEndPosition (enumeration of type string)	The end position of the chat invitation. Valid values include: • `TopLeft` • `Top` • `TopRight` • `Left` • `Center` • `Right` • `BottomLeft`

Field Name	Field Type	Description
		• `Bottom` • `BottomRight`
`inviteImage`	string	The custom button graphic that appears for the invitation.
`inviteStartPosition`	LiveChatButtonInviteStartPosition (enumeration of type string)	The start position of the chat invitation. Valid values include: • `TopLeft` • `TopLeftTop` • `Top` • `TopRightTop` • `TopRight` • `TopRightRight` • `Right` • `BottomRightRight` • `BottomRight` • `BottomRightBottom` • `Bottom` • `BottomLeftBottom` • `BottomLeft` • `BottomLeftLeft` • `Left` • `TopLeftLeft`
`isActive`	boolean	Specifies whether the chat button or invitation is active.
`label`	string	Specifies the text that appears on the button.
`numberOfReroutingAttempts`	int	Specifies the number of times a chat request can be rerouted to available agents if all agents reject the chat request. Available in API version 30.0 and later.
`offlineImage`	string	Specifies the image that appears on the button when no agents are available to chat.
`onlineImage`	string	Specifies the image that appears on the button when agents are available to chat.
`optionsCustomRoutingIsEnabled`	boolean	Indicates whether custom routing is enabled for incoming chat requests (`true`) or not (`false`). Available in API version 30.0 and later.

Field Name	Field Type	Description
optionsHasChasitorIdleTimeout	boolean	Indicates whether the visitor idle timeout feature is enabled. Available in API version 35.0 and later.
optionsHasInviteAfterAccept	boolean	Indicates whether a new chat invitation triggers after a customer accepts a previous chat invitation (`true`) or not (`false`).
optionsHasInviteAfterReject	boolean	Indicates whether a new chat invitation triggers after a customer rejects a previous chat invitation (`true`) or not (`false`).
optionsHasRerouteDeclinedRequest	boolean	Indicates whether a chat request, which has been rejected by all available agents, is rerouted to available agents again (`true`) or not (`false`). Available in API version 30.0 and later.
optionsIsAutoAccept	boolean	Indicates whether a chat request is automatically accepted by the agent it's assigned to (`true`) or not (`false`). For chat buttons and automated chat invitations with `routingType` set to `MostAvailable` or `LeastActive`. Available in API version 30.0 and later.
optionsIsInviteAutoRemove	boolean	Indicates whether a chat invitation is set to automatically disappear from a customer's screen after a certain amount of time (`true`) or not (`false`).
overallQueueLength	int	Specifies the maximum number of chat requests that are allowed to queue.
perAgentQueueLength	int	Specifies the number of chat requests that are allowed to queue for an agent with the required skills.
postChatPage	string	Specifies the name of the post-chat form to which customers are routed when the chat ends.
postChatUrl	string	Specifies the URL of the post-chat form to which customers are routed when the chat ends.
preChatFormPage	string	Specifies the name of the pre-chat form to which customers are routed before a chat begins.
preChatFormUrl	string	Specifies the URL of the pre-chat form to which customers are routed when the chat begins.
pushTimeOut	int	Specifies the number of seconds an agent has to answer an incoming chat request before the request is routed to another agent.

Field Name	Field Type	Description
routingType	LiveChatButtonRoutingType (enumeration of type string)	Specifies how incoming chats are routed to agents when a customer pushes a button. Valid values are: • Choice • LeastActive • MostAvailable
site	string	Specifies the Salesforce site that hosts your custom chat button images or custom chat page. You must have the CustomDomain permission enabled in your organization before you can use a Salesforce site with Chat.
skills	LiveChatButtonSkills	Specifies the skills associated with the button. When a customer clicks the button to chat, they're automatically routed to agents with those skills.
timeToRemoveInvite	int	Specifies how long the invitation is displayed (in seconds) to customers before it disappears.
type	LiveChatButtonType (enumeration of type string)	Required. The chat button type. Valid values are: • Standard • Invite
windowLanguage	Language	Specifies the language preferences for the chat window associated with the button.

LiveChatButtonSkills

Represents the skills associated with a chat button or invitation.

Fields

Field Name	Field Type	Description
skill	string	Specifies the name of the skill.

LiveChatButtonDeployments

Represents the deployments associated with a chat button or invitation.

Fields

Field Name	Field Type	Description
deployment	string	Specifies the name of the deployment.

Declarative Metadata Sample Definition

Here's a sample of a `liveChatButton` file.

```
<?xml version="1.0" encoding="UTF-8"?>
<LiveChatButton xmlns="http://soap.sforce.com/2006/04/metadata">
    <deployments/>
    <enableQueue>false</enableQueue>
    <isActive>true</isActive>
    <label>CustomerSupportButton</label>
    <optionsCustomRoutingIsEnabled>false</optionsCustomRoutingIsEnabled>
    <optionsHasChasitorIdleTimeout>false</optionsHasChasitorIdleTimeout>
    <optionsHasInviteAfterAccept>false</optionsHasInviteAfterAccept>
    <optionsHasInviteAfterReject>false</optionsHasInviteAfterReject>
    <optionsHasRerouteDeclinedRequest>false</optionsHasRerouteDeclinedRequest>
    <optionsIsAutoAccept>false</optionsIsAutoAccept>
    <optionsIsInviteAutoRemove>false</optionsIsInviteAutoRemove>
    <postChatUrl>https://help.salesforce.com</postChatUrl>
    <routingType>Choice</routingType>
    <skills>
        <skill>Chat</skill>
    </skills>
    <type>Standard</type>
</LiveChatButton>
```

Note: If you update your chat button through the Metadata API, be sure to update all Web pages that use the same chat button code.

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

LiveChatDeployment

Represents the configuration settings for a specific Chat deployment, such as the branding image for the deployment and whether or not chat transcripts are automatically saved.

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

LiveChatDeployment values are stored in the `<developer_name>.liveChatDeployment` file in the `liveChatDeployments` directory.

Version

LiveChatDeployment is available in API version 28.0 and later.

Fields

Field Name	Field Type	Description
`brandingImage`	string	Specifies the branding image for the deployment.
`connectionTimeoutDuration`	int	Indicates the amount of time before the chat times out, in seconds.
`ConnectionWarningDuration`	int	Indicates the amount of time before a time-out warning is displayed to the agent, in seconds.
`displayQueuePosition`	boolean	(Pilot) Determines whether a customer's queue position is displayed in a standard chat window while the customer waits for an agent to respond to the chat request (`true`) or not (`false`). This field is available as a pilot in API version 32.0. To enable this field, contact Salesforce.
`domainWhiteList`	LiveChatDeploymentDomainWhiteList	Specifies the list of domains that can host the deployment.
`enablePrechatApi`	boolean	Indicates whether or not the pre-chat API is enabled for the deployment (`true`) or not (`false`).
`enableTranscriptSave`	boolean	Indicates whether chat transcripts are automatically saved after a chat ends (`true`) or not (`false`).
`label`	string	Specifies the name of the deployment.
`mobileBrandingImage`	string	Specifies the branding image for the deployment that appears when customers access the deployment on a mobile device.
`site`	string	Specifies the site that hosts the images for the deployment. Note: You must have the CustomDomain permission enabled in your organization before you can use a Salesforce site with Chat.
`windowTitle`	string	Specifies the title of the window associated with the deployment.

LiveChatDeploymentDomainWhiteList

Represents a Chat deployment's domain whitelist.

Fields

Field Name	Field Type	Description
`domain`	string	Specifies a domain that can host the deployment.

Declarative Metadata Sample Definition

This is a sample of a `liveChatDeployment` file.

```
<?xml version="1.0" encoding="UTF-8"?>
<LiveChatDeployment xmlns="http://soap.sforce.com/2006/04/metadata">
    <label>My Deployment 1</label>
    <brandingImage>pkb_image_bannerBg</brandingImage>
    <mobileBrandingImage>pkb_image_bgBottom</mobileBrandingImage>
    <domainWhiteList>
        <domain>mydomain</domain>
        <domain>test</domain>
    </domainWhiteList>
    <enableTranscriptSave>true</enableTranscriptSave>
    <site>GL_Knowledge_Base</site>
    <windowTitle>My window title</windowTitle>
</LiveChatDeployment>
```

Note: If you update your deployment through the Metadata API, be sure to update all Web pages that use the same deployment code.

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

LiveChatSensitiveDataRule

Represents a rule for masking or deleting data of a specified pattern. Written as a regular expression (regex).

Use this object to mask or delete data of specified patterns, such as credit card, social security, phone and account numbers, or even profanity. This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

LiveChatSensitiveDataRule components have the suffix `.liveChatSensitiveDataRule` and are stored in the `liveChatSensitiveDataRule` folder.

Version

LiveChatSensitiveDataRule components are available in API version 35.0 and later.

Fields

Field Name	Field Type	Description
`actionType`	SensitiveDataActionType (enumeration of type string)	Required. The action to take on the text when the sensitive data rule is triggered. Possbile values are: • `Remove` • `Replace`
`description`	string	The description of the sensitive data rule—for example, "Block social security numbers."
`enforceOn`	int	Required. Determines the roles on which the rule is enforced. The value is determined using bitwise OR operation. There are seven possible values: **1.** Rule enforced on Agent **2.** Rule enforced on Visitor **3.** Rule enforced on Agent and Visitor **4.** Rule enforced on Supervisor **5.** Rule enforced on Agent and Supervisor **6.** Rule enforced on Visitor and Supervisor **7.** Rule enforced on Agent, Visitor, and Supervisor
`isEnabled`	boolean	Required. Specifies whether a sensitive data rule is active (`true`) or not (`false`). Default value (if none is provided) is `false`.
`pattern`	string	Required. The pattern of text blocked by the rule. Written as a JavaScript regular expression (regex).
`replacement`	string	The string of characters that replaces the blocked text (if `ActionType` *`Replace`* is selected).

Declarative Metadata Sample Definition

The following is an example of a LiveChatSensitiveDataRule component.

```
<LiveChatSensitiveDataRule xmlns="http://soap.sforce.com/2006/04/metadata">
    <actionType>REPLACE</actionType>
    <enforceOn>7</enforceOn>
    <isEnabled>true</isEnabled>
    <pattern>[aeiou]</pattern>
    <replacement>œ</replacement>
</LiveChatSensitiveDataRule>
```

The following is an example `package.xml` that references the previous definition.

```
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <!-- To be used from
support.liveagent.testsuite.unifiedouting.testDeployButtonMDAPIWithExistingQueue -->
    <apiAccessLevel>Unrestricted</apiAccessLevel>

    <types>
        <members>Change_For_all</members>
        <name>LiveChatSensitiveDataRule</name>
    </types>

    <version>35.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

LoyaltyProgramSetup

Represents the configuration of a loyalty program process including its parameters and rules. Program processes determine how new transaction journals are processed. When new transaction journals meet the criteria and conditions for a program process, actions that are set up in the process are triggered for the transaction journals.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

LoyaltyProgramSetup components have the suffix `loyaltyProgramSetup` and are stored in the `loyaltyProgramSetups` folder.

Version

LoyaltyProgramSetup components are available in API version 54.0 and later for Loyalty Management and in API version 59.0 and later for Referral Marketing.

Special Access Rules

To use this metadata type, your org must have either B2C - Loyalty, B2C - Loyalty Plus, Loyalty Management - Growth, Loyalty Management - Advanced, or Referral Marketing license enabled.

Fields

Field Name	Description
`label`	**Field Type** string **Description** Name of the loyalty program that the program process is associated with. If a loyalty program or referral program with the specified name doesn't exist, a new LoyaltyProgram record is created. The name of a program must contain at least one alphanumeric character.
`programProcesses`	**Field Type** LoyaltyProgramProcess[] on page 1393 **Description** Collection of loyalty program processes associated with a loyalty program or a referral program.

LoyaltyProgramProcess

Represents a collection of fields relating to a loyalty program process.

Field Name	Description
`description`	**Field Type** string **Description** The description of the loyalty program process.
`executionType`	**Field Type** LoyaltyPgmProcExecutionType (enumeration of type string) **Description** The mode of processing transaction journals by the loyalty program process. Possible values are: • `Batch` • `BatchAndRealTime` • `RealTime`
`journalSubType`	**Field Type** string **Description** The subtype of transaction journals processed by the loyalty program process.

Field Name	Description
`journalType`	**Field Type** string **Description** The type of transaction journal processed by the loyalty program process. Possible values for loyalty program: • `Accrual` • `Redemption` Possible value for referral program: • `Referral`
`loyaltyTierGroup`	**Field Type** string **Description** The tier group of a loyalty program. This field is available in API version 56.0 and later. This field isn't applicable for referral programs.
`parameters`	**Field Type** LoyaltyProgramProcessParameter[] on page 1395 **Description** The parameters associated with the loyalty program process.
`processName`	**Field Type** string **Description** Required. The name of the loyalty program process.
`processType`	**Field Type** string **Description** Required. The type of records processed by the loyalty program process. For referral programs, the process type is `TransactionJournal`.
`rules`	**Field Type** LoyaltyProgramProcessRule[] on page 1399 **Description** The rules associated with the loyalty program process.

Field Name	Description
`status`	**Field Type** LoyaltyPgmProcStatus (enumeration of type string) **Description** The status of the loyalty program process. Possible values are: • `Active` • `Draft` • `Inactive` Note: Only active program processes can process transaction journals.

LoyaltyProgramProcessParameter

Represents a collection of fields relating to a parameter that's associated with the program process. Parameters are dynamic or fixed values that are used in rule. You can define the value of a parameter based on its type and data type.

Field Name	Description
`condition`	**Field Type** LoyaltyProgramProcessCondition on page 1397 **Description** The filter condition that decides which records are stored in the parameter.
`dataType`	**Field Type** LoyaltyPgmProcParmDataType (enumeration of type string) **Description** The data type of the parameter. Determines the type of value that can be stored in the parameter. Possible values are: • `Boolean` • `Date` • `DateTime` • `Numeric` • `Sobject` • `Text`
`decimalPlaces`	**Field Type** int **Description** The number of decimal places supported by the parameter when it is of the type Variable and data type Numeric.

Field Name	Description
`description`	**Field Type** string **Description** The description of the parameter.
`isCollection`	**Field Type** boolean **Description** Indicates whether the parameter can store multiple values when it is of the type Variable.
`isInput`	**Field Type** boolean **Description** Indicates whether a parameter can be used as an input outside the loyalty program process.
`isOutput`	**Field Type** boolean **Description** Indicates whether a parameter can be used as an output outside the loyalty program process.
`objectName`	**Field Type** string **Description** Name of the object whose records are stored by the parameter when it is of the type Variable and data type sObject.
`parameterName`	**Field Type** string **Description** Required. The name of the parameter.
`parameterType`	**Field Type** LoyaltyPgmProcParmType (enumeration of type string) **Description** The type of value the parameter can store. Possible values are: • `Constant`

Field Name	Description
	• `Formula` • `Variable`
`value`	**Field Type** string **Description** The value of the parameter when it is of the type Variable or Formula and isn't of the data type sObject.

LoyaltyProgramProcessCondition

Represents a collection of fields relating to a condition. Conditions filter records that parameters store or check whether child actions must be triggered for a transaction journal.

Field Name	Description
`conditionCriteria`	**Field Type** string **Description** Required. The criteria that determine when the condition is met by a record or by a transaction journal.
`conditionFilterCriteria`	**Field Type** LoyaltyProgramProcessConditionFilterCriteria[] on page 1397 **Description** The filter criteria that determines which records or transaction journals are filtered.
`conditionName`	**Field Type** string **Description** Required. The name of the condition.

LoyaltyProgramProcessConditionFilterCriteria

Represents a collection of fields relating to a filter criteria that's part of a condition. Multiple filter criteria can be added for a condition. Filter criteria determine which records are filtered by related condition.

Field Name	Description
`operator`	**Field Type** LoyaltyPgmProcCondOperator (enumeration of type string) **Description** Required. The operator of the filter criteria. Possible values are: • `Contains` • `DoesNotContain` • `EndsWith` • `Equals` • `GreaterThan` • `GreaterThanOrEquals` • `IsNotNull` • `IsNull` • `LessThan` • `LessThanOrEquals` • `NotEquals` • `StartsWith`
`sequence`	**Field Type** int **Description** Required. The sequence number of the filter criteria within a condition.
`sourceFieldName`	**Field Type** string **Description** Required. The name of the field used in the filter criteria.
`value`	**Field Type** string **Description** The value of the filter criteria.
`valueType`	**Field Type** LoyaltyPgmProcCondType (enumeration of type string)

Field Name	Description
	Description Required. The type of value specified in the filter criteria. Possible values are: • `Formula` • `Literal` • `Lookup` • `Parameter`

LoyaltyProgramProcessRule

Represents a collection of fields relating to a rule. A rule consists of a set of conditions and actions.

Field Name	Description
`actions`	**Field Type** LoyaltyProgramProcessAction[] on page 1400 **Description** The actions associated with the rule.
`conditions`	**Field Type** LoyaltyProgramProcessCondition[] on page 1397 **Description** The conditions associated with the rule.
`description`	**Field Type** string **Description** The description of the rule.
`endDate`	**Field Type** date **Description** The date until which the rule processes transaction journals.
`previousRule`	**Field Type** string **Description** The rule that processes new transaction journals before the current rule. The current rule is triggered when the previous rule completes processing transaction journals.

Field Name	Description
`promotion`	**Field Type** string **Description** The promotion associated with the rule. When a promotion is associated with a rule, the start date, end date, and status of the promotion determines the corresponding fields of the rule.
`ruleName`	**Field Type** string **Description** Required. The name of the rule.
`startDate`	**Field Type** date **Description** The date from which the rule starts processing transaction journals.
`status`	**Field Type** LoyaltyPgmProcRuleStatus (enumeration of type string) **Description** The status of the rule. Possible values are: • `Active` • `Draft` • `Inactive`
`stepMappings`	**Field Type** LoyaltyProgramProcessRuleStepMapping[] on page 1405 **Description** The list of step mappings associated with rule.

LoyaltyProgramProcessAction

Represents a collection of fields relating to an action.

Field Name	Description
`actionName`	**Field Type** string

Field Name	Description
	Description Required. The name of the action.
`actionParameters`	**Field Type** LoyaltyProgramProcessActionParameter[] on page 1404 **Description** The parameters of the action.
`actionType`	**Field Type** LoyaltyPgmProcActionType (enumeration of type string) **Description** Required. The type of action. Possible values are: • These values are used in Loyalty Management: – `AssignParameterValues`—Assigns values to parameters. – `AssignBadgeToMember`—Assigns a badge to a loyalty program member. This value is available in API version 56.0 and later. – `Crud`—Creates or updates records in the target object. This value is available in API version 56.0 and later. – `CheckMemberBadgeAssignment`—Checks whether a badge is assigned to a loyalty program member. This value is available in API version 56.0 and later. – `ChangeMemberTier`—Changes the tier of a loyalty program member. This value is available in API version 56.0 and later. – `CreditPoints`—Credits points to the loyalty program member associated with the transaction journal that's processed by the rule. – `DebitPoints`—Debits points from the points balance of the loyalty program member associated with the transaction journal that's processed by the rule. – `GetMemberAttributesValues`—Gets the details of the loyalty program member's attribute value for the selected engagement attribute. This value is available in API version 55.0 and later. – `GetMemberPointBalance`—Gets the points balance of a loyalty program member. – `GetMemberPromotions`—Get promotions of a loyalty program member. This value is available in API version 56.0 and later. – `GetMemberTier`—Gets the tier details of a loyalty program member. – `GetOutputsFromDecisionTable`—Gets outputs provided by a decision table. This value is available in API version 56.0 and later.

Field Name	Description
	– —Adds the specified value to the loyalty program member's usage towards achieving a cumulative promotion by a specified value. – `IncreaseUsageForCumulativePromotion`—Increases a loyalty program member's usage for a cumulative promotion. – `IssueVoucher`—Issues a voucher to the loyalty program member associated with the transaction journal that's processed by the rule. – `RedeemVoucher`—Redeems a voucher for the loyalty program member associated with the transaction journal that's processed by the rule. This value is available in API version 58.0 and later. – —Updates the loyalty program member's usage towards achieving a cumulative promotion by a specified value. – `RunFlow`—Runs a flow. – `RunProgramProcess`—Runs an active loyalty program process as a subprocess. This value is available in API version 56.0 and later. – `SendMail`—Sends emails to the loyalty program member for whom the process is run. This value is available in API version 59.0 and later. – `UpdateCurrentValueForMemberAttribute`—Updates the loyalty program member's current attribute value for the selected engagement attribute. This value is available in API version 55.0 and later. – `UpdatePointBalance`—Updates the points balance of the loyalty program member associated with the transaction journal that's processed by the rule. – `UpdateUsageForCumulativePromotion`—Updates a loyalty program member's usage for a cumulative promotion. • These values are used in Referral Marketing: – `AssignParameterValues`—Assigns values to parameters. – `Crud`—Creates or updates records in the target object. – `GetMemberAttributesValues`—Gets the details of an advocate's attribute value for the selected engagement attribute. – `GetMemberPromotions`—Gets the promotions of an advocate. – `GetOutputsFromDecisionTable`—Gets outputs provided by a decision table. – `IssueVoucher`—Issues a voucher to an advocate or a referred friend. – `RedeemVoucher`—Redeems a voucher for an advocate or a friend. – `SendMail`—Sends emails to a referral program's advocates and referrals. – `UpdateCurrentValueForMemberAttribute`—Updates an advocate's current attribute value for the selected engagement attribute.
`crudActionType`	**Field Type** LoyaltyPgmProcCrudActType (enumeration of type string)

Field Name	Description
	Description The type of operation to perform on target object records by the action. This field is available from API version 56.0 and later. Note: This field is required when the `actionType` field value is `CRUD`. Possible values are: • `create` • `update`
`decisionTable`	**Field Type** string **Description** The decision that's invoked by the action for the transaction journal that's processed by the rule.
`decisionTableDatasetLink`	**Field Type** string **Description** The dataset link associated with the selected decision table.
`entityApiName`	**Field Type** string **Description** The API name of the target object. This field is available from API version 56.0 and later. Note: This field is required when the `actionType` field value is `CRUD`.
`flowDefinition`	**Field Type** string **Description** The flow that's run by the action for the transaction journal that's processed by the rule. The selected flow must be of the type LoyaltyManagementFlow.
`loyaltyProgramProcess`	**Field Type** string **Description** The subprogram processes that's run by the action. This field is available from API version 56.0 and later. Note: This field is required when the `actionType` field value is `RunProgramProcess`.

LoyaltyProgramProcessActionParameter

Represents a collection of fields relating to an action parameter. A parameter is either an input or an output for the action. Input parameters store the values used by the action. Output parameters store the result of the action.

Field Name	Description
`operator`	**Field Type** LoyaltyPgmProcActParamOper (enumeration of type string) **Description** The type of operator used in the action. This field is available in API version 56.0 and later. Possible value is: • `Equals`
`parameterName`	**Field Type** string **Description** Required. The name of parameter. The parameter name must be the same as the input or the output field that's supported depending on the associated action's type.
`sequenceNumber`	**Field Type** int **Description** The sequence number of the parameter in the action. This field is available in API version 56.0 and later.
`value`	**Field Type** string **Description** Required. The value of the parameter.
`valueType`	**Field Type** LoyaltyPgmProcActParamType (enumeration of type string) **Description** The type of value to provide in the parameter. This field is available in API version 56.0 and later. Possible values are: • `Literal`—A constant value. • `Parameter`—A runtime value passed using a parameter.

LoyaltyProgramProcessRuleStepMapping

Represents a collection of fields relating to a step mapping. Map conditions with child actions or map an action without a parent step.

Field Name	Description
associatedStep	**Field Type** string **Description** Required. The action that's associated with the mapping.
parentStep	**Field Type** string **Description** The condition that contains one or more child actions.
sequence	**Field Type** int **Description** Required. The sequence number of the mapping within a rule.

Declarative Metadata Sample Definition

The following is an example of a LoyaltyProgramSetup component.

```
<?xml version="1.0" encoding="UTF-8"?>
<LoyaltyProgramSetup xmlns="http://soap.sforce.com/2006/04/metadata">
    <label>Cloud Kicks Inner Circle</label>
    <programProcesses>
        <executionType>RealTime</executionType>
        <parameters>
            <dataType>Numeric</dataType>
            <decimalPlaces>0</decimalPlaces>
            <isCollection>false</isCollection>
            <isInput>false</isInput>
            <isOutput>false</isOutput>
            <parameterName>VoucherValue</parameterName>
            <parameterType>Constant</parameterType>
            <value>50</value>
        </parameters>
        <processName>Issue Vouchers</processName>
        <processType>Transaction Journal</processType>
        <rules>
            <actions>
                <actionName>Issue High Transaction Value Voucher</actionName>
```

```
        <actionParameters>
            <operator>Equals</operator>
            <parameterName>VoucherDefinitionName</parameterName>
            <sequenceNumber>1</sequenceNumber>
            <value>Voucher for High Value Transactions</value>
            <valueType>Literal</valueType>
        </actionParameters>
        <actionParameters>
            <operator>Equals</operator>
            <parameterName>VoucherCode</parameterName>
            <sequenceNumber>2</sequenceNumber>
            <value>{!TransactionJournal.Order.Id}</value>
        </actionParameters>
        <actionParameters>
            <operator>Equals</operator>
            <parameterName>VoucherEffectiveDate</parameterName>
            <sequenceNumber>3</sequenceNumber>
            <value>DATEVALUE(&amp;quot;2021-11-21 00:00:00&amp;quot;)</value>
        </actionParameters>
        <actionParameters>
            <operator>Equals</operator>
            <parameterName>VoucherExpirationDate</parameterName>
            <sequenceNumber>4</sequenceNumber>
            <value>DATEVALUE(&amp;quot;2022-01-01 00:00:00&amp;quot;)</value>
        </actionParameters>
        <actionParameters>
            <operator>Equals</operator>
            <parameterName>VoucherFaceValue</parameterName>
            <sequenceNumber>5</sequenceNumber>
            <value>{!VoucherValue}</value>
        </actionParameters>
        <actionType>IssueVoucher</actionType>
    </actions>
    <conditions>
        <conditionCriteria>1</conditionCriteria>
        <conditionFilterCriteria>
            <operator>GreaterThanOrEquals</operator>
            <sequence>1</sequence>
          <sourceFieldName>TransactionJournal.TransactionAmount</sourceFieldName>

            <value>100</value>
            <valueType>Literal</valueType>
        </conditionFilterCriteria>
        <conditionName>New Condition</conditionName>
        <conditionType>Condition</conditionType>
    </conditions>
    <endDate>2022-01-01</endDate>
    <ruleName>Issue Voucher for Transactions Above $100</ruleName>
    <startDate>2021-11-21</startDate>
    <status>Draft</status>
    <stepMappings>
        <associatedStep>New Condition</associatedStep>
        <sequence>1</sequence>
    </stepMappings>
```

```
            <stepMappings>
                <associatedStep>Issue High Transaction Value Voucher</associatedStep>
                <parentStep>New Condition</parentStep>
                <sequence>1</sequence>
            </stepMappings>
        </rules>
        <status>Draft</status>
    </programProcesses>
</LoyaltyProgramSetup>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
   ~ Copyright 2020 Salesforce, Inc.
   ~ All Rights Reserved
   ~ Company Confidential
-->
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
   <types>
      <members>*</members>
      <name>LoyaltyProgramSetup</name>
   </types>
   <version>54.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ManagedContentType

Represents the definition of custom content types for use with Salesforce CMS. Custom content types are displayed as forms with defined fields.

This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

ManagedContentType components have the suffix `managedContentType` and are stored in the `managedContentTypes` folder.

Version

ManagedContentType components are available in API version 47.0 and later.

Special Access Rules

ManagedContentType is only available if Salesforce CMS and digital experiences are enabled for your org.

Fields

Field Name	Field Type	Description
`description`	string	Describes the custom content type defined in this ManagedContentType declaration.
`developerName`	string	Required. Unique name for the custom content type. For example: *`OurSpecialContent_c`*
`isMetadataContent`	boolean	When `true`, any content created from this content type is converted to metadata. Default value is `false`. Availabe in API version 63.0 and later.
`managedContentNodeTypes`	ManagedContentNodeType[]	Nodes included as part of this custom content type. When rendered as a form in the Digital Experiences app, each node is represented as an individual field.
`masterLabel`	string	Required. Declares the name of the content type as it appears in the UI.

ManagedContentNodeType

Represents the structure of individual nodes within the custom content type.

Field Name	Field Type	Description
`helpText`	string	Provides assistive text in the UI, displayed as an info bubble for the field. If this field is empty, no info bubble icon or text is displayed. For example: <IMG?>
`isLocalizable`	boolean	Declares a field as localizable and consumable by <loc MDAPI reference> (`true`) or not (`false`). Default is `false`. Note: NodeTypes `IMG`, `URL`, `DATE`, and `DATETIME` can't be localized.
`isRequired`	boolean	Declares a field as required (`true`) or not (`false`). Fields declared as required are indicated by a red asterisk. If a value isn't added to the field in the custom content type, the form can't be saved and a standard error is displayed. Default is `false`. Note: When `nodeType on page 1409` is set to `NAMEFIELD` on a field, `isRequired` must also be set to `True` for that field.

Field Name	Field Type	Description
`nodeLabel`	string	Required. Declares the label for the field as it appears in the UI. In enhanced workspaces, the system generates a Title field by default. To prevent having multiple Title fields on the UI when you create a custom content type for use in an enhanced workspace, don't use Title as the label for `nodeLabel`.
`nodeName`	string	Required. Unique name of the `nodeType` within the content type. `nodeName` is a simple text field that allows up to 100 alphanumeric characters and underscores. The name must begin with a letter, not include spaces, can't have two consecutive underscores, and can't end with an underscore. In enhanced workspaces, the system generates a Title field by default. To prevent having multiple Title fields on the UI when you create a custom content type for use in an enhanced workspace, don't use Title as the label for `nodeName`.
`nodeType`	MCNodeType (enumeration of type string)	Required. Identifies the supported type of content in the node. Passed as a string. There's a maximum of 15 node types per content type. Values are case insensitive but are returned in all capital letters as shown. Valid values are: • `TEXT` Simple text node (max length=255 characters) • `MTEXT` Multi-line text node (max length=2000 characters) • `RTE` Rich text node (max length=65536 characters) • `IMG` Image node Note: `IMG` node types can't be localized. Set `isLocalizable` to false for images. • `URL` URL node (max length=255 characters) Note: `URL` accepts protocol string values starting with http://, https://, mailto:, tel:, and /. Note: `URL` node types can't be localized. Set `isLocalizable` to false for URLs. • `DATE` Date node Note: `DATE` accepts dates only in the format yyyy-MM-dd.

Field Name	Field Type	Description
		Note: `DATE` node types can't be localized. Set `isLocalizable` to false for dates. • `DATETIME` Datetime node Note: `DATETIME` accepts date and time in the format: yyyy-MM-dd'T'HH:mm:ss.SSS'Z' (UTC datetime in ISO 8601 format). Note: `DATETIME` node types can't be localized. Set `isLocalizable` to false for datetime notes. • `NAMEFIELD` Note: `NAMEFIELD` declares the field as the name that represents the content when referenced in the UI. For example, text entered in this field displays as a list of available content in the Digital Experiences app or as a piece of content available for inclusion in a collection in an Experience Cloud site. One, and only one, `nodeType` in your managed content type must be declared as `NAMEFIELD`. `NAMEFIELD` is a string of 200 characters or fewer. In enhanced workspaces, the system generates a Title field by default. To prevent having multiple Title fields on the UI when you create a custom content type for use in an enhanced workspace, don't use Title as the label for `nodeName` or `nodeLabel` for the `NAMEFIELD` node. If you've already named `nodeName` Title, choose a different label for `nodeLabel` to prevent confusion on the content creation page. When `NAMEFIELD` is used, `isRequired` must also be set to `true` for the field.
`placeholderText`	string	Provides assistive text in the UI, displayed as placeholder, or ghost text, in a field before any entry is made. For example, `Enter a title for your article...`

Declarative Metadata Sample Definition

The following is an example of a ManagedContentType component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ManagedContentType xmlns="http://soap.sforce.com/2006/04/metadata">
  <developerName>myContentType</developerName>
 <masterLabel>My Content Type</masterLabel>
 <description>This is the description for my content type</description>
 <managedContentNodeTypes>
  <nodeName>title</nodeName>
```

```
  <nodeLabel>Content Title</nodeLabel>
  <nodeType>NAMEFIELD</nodeType>
  <placeholderText>Placeholder Text for title</placeholderText>
  <helpText>Help Text for title</helpText>
  <isLocalizable>true</isLocalizable>
  <isRequired>true</isRequired>
 </managedContentNodeTypes>
 <managedContentNodeTypes>
  <nodeName>textnode</nodeName>
  <nodeLabel>Content Text</nodeLabel>
  <nodeType>TEXT</nodeType>
  <placeholderText>Placeholder Text for Content Text</placeholderText>
  <helpText>Help Text for Content Text</helpText>
  <isLocalizable>true</isLocalizable>
  <isRequired>false</isRequired>
 </managedContentNodeTypes>
 <managedContentNodeTypes>
  <nodeName>richtextnode</nodeName>
  <nodeLabel>Content RichText</nodeLabel>
  <nodeType>RTE</nodeType>

 </managedContentNodeTypes>
 <managedContentNodeTypes>
  <nodeName>multilinetextnode</nodeName>
  <nodeLabel>Content MultilineText</nodeLabel>
  <nodeType>MTEXT</nodeType>
 </managedContentNodeTypes>
 <managedContentNodeTypes>
  <nodeName>imagenode</nodeName>
  <nodeLabel>Content Image</nodeLabel>
  <nodeType>IMG</nodeType>

 </managedContentNodeTypes>
</ManagedContentType>
```

Usage

For each custom content type you create, there must also be a CMS Content page created in any Experience Cloud site that displays the content. Each Content page serves as the detail page for all content of a single content type. See Create Custom Pages with Experience Builder.

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ManagedEventSubscription (Beta)

Represents a managed event subscription in Pub/Sub API. Use a managed event subscription to track the events that a subscriber client consumed and resume a subscription where it left off. This type extends the metadata type and inherits its `fullName` field.

Note: This feature is a Beta Service. Customer may opt to try such Beta Service in its sole discretion. Any use of the Beta Service is subject to the applicable Beta Services Terms provided at Agreements and Terms.

File Suffix and Directory Location

ManagedEventSubscription components have the suffix `.managedEventSubscription` and are stored in the `managedEventSubscriptions` folder.

Version

ManagedEventSubscription components are available in API version 60.0 and later.

Special Access Rules

You must have the Customize Application permission to deploy and retrieve this type.

Fields

Field Name	Field Type	Description
`defaultReplay`	EventSubscriptionReplayPreset (enumeration of type string)	The position in the stream where the subscription starts when the client initiates the subscription for the first time or if the client doesn't commit a Replay ID. Possible values are: • `LATEST`—(Default) The subscription starts from the latest events received. This option skips sending events that were published when the client was disconnected. • `EARLIEST`—The subscription starts from the earliest events stored in the event bus. This option sends new events and any other events less than 72 hours old. You can reprocess all stored events and catch up on missed events. Use this option sparingly. Subscribing with the `EARLIEST` option when a large number of event messages are stored can slow performance and exhaust the event delivery allocation.
`errorRecoveryReplay`	EventSubscriptionReplayPreset (enumeration of type string)	The position in the stream where the subscription restarts if the committed Replay ID is invalid. The Replay ID can be invalid if it's older than the event retention window. Possible values are: • `LATEST`—(Default) The subscription restarts from the latest events received. This option skips sending events that were published when the client was disconnected. • `EARLIEST`—The subscription restarts from the earliest events stored in the event bus. This option sends new events and any other events less than 72 hours old. You can reprocess all stored events and catch up on missed events. Use this option sparingly. Subscribing with the `EARLIEST` option when a large number of event

Field Name	Field Type	Description
		messages are stored can slow performance and exhaust the event delivery allocation.
`label`	string	The label for the managed subscription.
`state`	EventSubscriptionAdminState (enumeration of type string)	The execution state that the `ManagedSubscribe` RPC call consumes. If `state` is set to `RUN`, the subscription starts when the `ManagedSubscribe` RPC call is made. Otherwise, the subscription doesn't start. If an administrator later changes `state` from `RUN` to `STOP`, the system notifies the Pub/Sub API client of the new `state` value and the subscription disconnects. Also, the stored Replay ID value that was committed previously is deleted. The next time the `ManagedSubscribe` RPC call is made after `state` is changed from `STOP` to `RUN`, the subscription starts from the `defaultReplay` value. The possible values for `state` are: • `RUN`—(Default) The subscription is running and delivering new events to the Pub/Sub API client. • `STOP`—The subscription is stopped. No events are delivered to the Pub/Sub API client during this state and the previously committed Replay ID is deleted. • `PAUSE`—Reserved for internal use.
`topicName`	string	The topic name of the platform event or change event or the channel name of a custom platform event channel or custom or standard change data capture channel. The topic name can be one of the following values. • For a platform event—`/event/EventName__e` • For a custom platform event channel—`/event/CustomPEChannel__chn` • For the standard change event channel—`/data/ChangeEvents` • For a change event (replace `Object` with the object name)—`/data/ObjectChangeEvent`. For example, for Account, it's `/data/AccountChangeEvent`. • For a custom change event channel—`/data/CustomChangeChannel__chn`
`version`	string	Reserved for internal use.

Declarative Metadata Sample Definition

The following is an example of a ManagedEventSubscription component with the file name `My_Managed_Subscription.managedSubscription`.

```
<?xml version="1.0" encoding="UTF-8"?>
<ManagedEventSubscription xmlns="http://soap.sforce.com/2006/04/metadata">
    <defaultReplay>LATEST</defaultReplay>
    <errorRecoveryReplay>LATEST</errorRecoveryReplay>
    <label>My Managed Subscription</label>
    <state>RUN</state>
    <topicName>/event/Order_Event__e</topicName>
</ManagedEventSubscription>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>My_Managed_Subscription</members>
        <name>ManagedEventSubscription</name>
    </types>
    <version>64.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ManagedTopics

Represents navigational and featured topics managed in an Experience Cloud site.

Note: The related Experience Cloud site must exist before you deploy managed topics. (This occurs automatically when deploying an entire org.)

File Suffix and Directory Location

Components have the suffix managedTopics and are stored in the managedTopics folder. In that folder, you find separate files for each Experience Cloud site (for example, `SiteNameA.managedTopics` and `SiteNameB.managedTopics`).

Version

ManagedTopics components are available in API version 32.0 and later.

Fields

Field Name	Field Type	Description
`ManagedTopic`	ManagedTopic	Represents a specific navigational or featured topic.

ManagedTopic

Field Name	Field Type	Description
`name`	string	The topic name.
`managedTopicType`	string	The topic type: “Navigational” or “Featured”
`topicDescription`	string	An optional description of topic contents. This field is accessible only via the API; there is no corollary in the user interface.
`parentName`	string	The name of a parent topic for which this topic is a child. Child topics are accessible from the subtopics section of the parent topic page and their feeds are added to the parent topic feed. Only navigational topics support parent-child relationships.
`position`	int	The placement of this topic relative to others of the same type. The results differ depending on topic type: • For top-level navigational topics, `position` arranges the Topics menu in the Experience Cloud site. • For child navigational topics, it arranges sibling topics in the subtopics section. • For featured topics, it arranges topic thumbnail images on the Experience Cloud site home page. Enter a number between 0 and 24. (The maximum amount of navigational or featured topics is 25.)

Declarative Metadata Sample Definition

The following example retrieves or deploys managed topics for all sites:

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>ManagedTopics</name>
    </types>
    <version>32.0</version>
</Package>
```

The following example shows a package.xml file referencing the ManagedTopics component:

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>SiteName</members>
        <name>ManagedTopics</name>
    </types>
    <version>32.0</version>
</Package>
```

The following example shows the ManagedTopics component itself:

```
<?xml version="1.0" encoding="UTF-8"?>
<ManagedTopics>
    <ManagedTopic>
        <name>Running</name>
        <managedTopicType>Navigational</managedTopicType>
        <topicDescription>Training advice</topicDescription>
        <parentName></parentName>
        <position>0</position>
    </ManagedTopic>
    <ManagedTopic>
        <name>Hiking</name>
        <managedTopicType>Navigational</managedTopicType>
        <topicDescription>Routes and gear</topicDescription>
        <parentName></parentName>
        <position>1</position>
    </ManagedTopic>
        <ManagedTopic>
            <name>Trails</name>
            <managedTopicType>Navigational</managedTopicType>
            <topicDescription>Maps for local favorites</topicDescription>
            <parentName>Hiking</parentName>
            <position>0</position>
        </ManagedTopic>
        <ManagedTopic>
            <name>Backpacks</name>
            <managedTopicType>Navigational</managedTopicType>
            <topicDescription>Recommended models</topicDescription>
            <parentName>Hiking</parentName>
            <position>1</position>
        </ManagedTopic>
    <ManagedTopic>
        <name>Footwear</name>
        <managedTopicType>Featured</managedTopicType>
        <topicDescription>Suggested types for each sport</topicDescription>
        <parentName></parentName>
        <position>0</position>
    </ManagedTopic>
    <ManagedTopic>
        <name>Conditioning</name>
        <managedTopicType>Featured</managedTopicType>
        <topicDescription>How to get fit for any activity</topicDescription>
        <parentName></parentName>
```

```
        <position>1</position>
    </ManagedTopic>
</ManagedTopics>
```

Usage

Managed topic images that are uploaded in API version 50.0 and later are stored as asset files. To migrate managed topic images that are uploaded in API version 50.0 and later, use the ContentAsset metadata type. To migrate managed topic images that were uploaded in API version 49.0 and earlier, use the Document metadata type.

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

MarketingAppExtension

Represents an integration with a third-party app or service that is used to work with prospects.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata type and inherits its `fullName` field.

File Suffix and Directory Location

MarketingAppExtension components have the suffix `.marketingappextension` and are stored in the `marketingappextensions` folder.

Version

MarketingAppExtension components are available in API version 54.0 and later.

Special Access Rules

The first Salesforce or designated marketing admin to access Marketing App Extensions in an org must have the Manage Public List Views user permission. Subsequent users don't need the permission to work with the feature.

Fields

Field Name	Description
`description`	**Field Type** string **Description** The description of the extension for internal reference. Appears in the UI.
`isActive`	**Field Type** boolean **Description** This field makes data for a Marketing App Extension available to use in Account Engagement automations. Label is Active in Automations. The default value is `false`. Appears in the UI.
`isProtected`	**Field Type** boolean
`marketingAppExtActions`	**Field Type** MarketingAppExtAction on page 1421[] **Description** This field is a related list of associated external actions.
`marketingAppExtActivities`	**Field Type** MarketingAppExtActivity on page 1418[] **Description** This field is a related list of associated external prospect activities.
`masterLabel`	**Field Type** string **Description** Required. Label for the MarketingAppExtension. In the UI, this field is Extension Name.

MarketingAppExtActivity

Represents an Activity Type, which is a prospect activity that occurs in a third-party app and can be used in Account Engagement automations.

Field Name	Description
`description`	**Field Type** string

Field Name	Description
	Description The description of the activity for internal reference. Appears in the UI.
`endpointUrl`	**Field Type** string **Description** A sample endpoint that can be used to help connect the activity type to a third-party app. Appears in the UI.
`isActive`	**Field Type** boolean **Description** This field makes data for the Activity Type available to use in Account Engagement automations. Label is Active in Automations. The default value is `false`. Appears in the UI.
`isProtected`	**Field Type** boolean
`marketingAppExtension`	**Field Type** string **Description** Required. The Marketing App Extension associated with the activity.
`masterLabel`	**Type** string **Description** Required. Label for the MarketingAppExtActivity. In the UI, this field is Activity Name.

Declarative Metadata Sample Definition

This example retrieves all Activity Types associated with the MarketingAppExtension component.

```
<?xml version="1.0" encoding="UTF-8"?>
<MarketingAppExtension xmlns="http://soap.sforce.com/2006/04/metadata">
    <description>VidLand extension for US region</description>
    <isActive>true</isActive>
    <marketingAppExtActivities>
        <fullName>user_attended</fullName>
        <description>User attended activity capture for VidLand</description>
        <isActive>true</isActive>
        <marketingAppExtension>VidLand_US</marketingAppExtension>
        <masterLabel>user attended</masterLabel>
    </marketingAppExtActivities>
```

```
    <marketingAppExtActivities>
        <fullName>user_registered</fullName>
        <description>User registered activity capture for VidLand</description>
        <isActive>true</isActive>
        <marketingAppExtension>VidLand_US</marketingAppExtension>
        <masterLabel>user registered</masterLabel>
    </marketingAppExtActivities>
    <masterLabel>VidLand_US</masterLabel>
</MarketingAppExtension>
```

This example `package.xml` references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
~ Copyright 2021 Salesforce, Inc.
~ All Rights Reserved
~ Company Confidential
-->
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
<types>
<members>VidLand_US</members>
<name>MarketingAppExtension</name>
</types>
</Package>
```

This example retrieves a specific Activity Type from the associated MarketingAppExtension component.

```
<?xml version="1.0" encoding="UTF-8"?>
<MarketingAppExtension xmlns="http://soap.sforce.com/2006/04/metadata">
    <description>VidLand extension for US region</description>
    <isActive>true</isActive>
    <marketingAppExtActivities>
        <fullName>user_attended</fullName>
        <description>User attended activity capture for VidLand</description>
        <isActive>true</isActive>
        <marketingAppExtension>VidLand_US</marketingAppExtension>
        <masterLabel>user attended</masterLabel>
    </marketingAppExtActivities>
    <masterLabel>VidLand_US</masterLabel>
</MarketingAppExtension>
```

This example `package.xml` references the previous definition.

```
<<?xml version="1.0" encoding="UTF-8"?>
<!--
~ Copyright 2021 salesforce.com, inc.
~ All Rights Reserved
~ Company Confidential
-->
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
<types>
<members>VidLand_US.user_attended</members>
<name>MarketingAppExtActivity</name>
</types>
<types>
<members>VidLand_US</members>
```

```
<name>MarketingAppExtension</name>
</types>
</Package>
```

MarketingAppExtAction

Represents an Action Type, which is an action that executes in a third-party app and can be used in Engagement Studio programs.

Field Name	Description
actionName	**Field Type** string **Description** The name of the action for internal use. Appears in the UI.
actionParams	**Field Type** string **Description** The parameters for the invocable action. Appears in the UI.
actionSchema	**Field Type** string **Description** The JSON schema for the invocable action. Appears in the UI.
actionSelector	**Type** string **Description** Invocable action selector. Appears in the UI.
apiName	**Field Type** string **Description** This name can contain only underscores and alphanumeric characters, and must be unique in your org. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. This field is automatically generated, but you can supply your own value if you create the record using the API. Appears in the UI.
Description	**Field Type** string **Description** The description of the action for internal reference. Appears in the UI.

Field Name	Description
isActive	**Field Type** boolean **Description** This field makes data for the Action Type available to use in Engagement Studio Label is Active in Automations. The default value is `false`. Appears in the UI.
isProtected	**Field Type** boolean
marketingAppExtension	**Field Type** string **Description** Required. The Marketing App Extension associated with the action.

Declarative Metadata Sample Definition

This example retrieves a specific action associated the MarketingAppExtension component.

```
<?xml version="1.0" encoding="UTF-8"?>
<MarketingAppExtension xmlns="http://soap.sforce.com/2006/04/metadata">
    <fullName>VidLand_US</fullName>
    <description>VidLand extension for US region</description>
    <isActive>true</isActive>
    <marketingAppExtActions>
        <marketingAppExtension>VidLand_US</marketingAppExtension>
        <apiName>register_user</apiName>
        <isActive>true</isActive>
        <description>Register User for VidLand</description>>
        <actionSelector>VidLand_Register_User</actionSelector>
        <actionSchema>
   <![CDATA[
       {
 "properties": {
   "UserId": {
 "type": "string",
 "title": ""
   },
   "WebinarId": {
 "type": "string",
 "value": "webinarIdXYZ"
   }
 },
 "view": {
   "components": [{
 "definition": "lightning/control",
 "scope": "#/properties/UserId"
```

```
    }]
  },
  "required": [
    "UserId",
    "WebinarId",
    "From",
    "Body"
  ]
   }
    ]]>
         </actionSchema>
         <actionParams>
         <![CDATA[
        {
  "isStandard": false,
     "type": "apex"
   }
    ]]>
         </actionParams>
         <actionName>Register User</actionName>
     </marketingAppExtActions>
     <masterLabel>VidLand US</masterLabel>
</MarketingAppExtension>
```

This example `package.xml` references the previous definition.

```
<<?xml version="1.0" encoding="UTF-8"?>
<!--
~ Copyright 2021 salesforce.com, inc.
~ All Rights Reserved
~ Company Confidential
-->
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
<types>
<members>VidLand_US</members>
<name>MarketingAppExtension</name>
</types>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

MatchingRule

Represents a matching rule that is used to identify duplicate records.

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

Matching rule components have the `.matchingRule` suffix and are stored in the `matchingRules` folder. The name of the component file is the standard or custom object name that is associated with the matching rule.

In API version 39.0 and later, MatchingRule supports the Person Account object.

- The component file name is `PersonAccount.matchingRule`.
- The component directory is `matchingRules`.

Version

MatchingRule is available in API version 33.0 and later.

Fields

Field Name	Field Type	Description
`booleanFilter`	string	Specifies filter logic conditions.
`description`	string	The description of the matching rule.
`label`	string	Required. The name of the matching rule.
`matchingRuleItems`	MatchingRuleItem	The criteria that make up a matching rule.
`ruleStatus`	MatchingRuleStatus (enumeration of type string)	Required. The activation status of the matching rule. Values are: • *`Inactive`* • *`Deactivating`* • *`DeactivationFailed`* • *`Active`* • *`Activating`* • *`ActivationFailed`* **Important:** The only valid values you can declare when deploying a package are *`Active`* and *`Inactive`*.

MatchingRuleItem

Field Name	Field Type	Description
`blankValueBehavior`	BlankValueBehavior (enumeration of type string)	Specifies how blank fields affect whether the fields being compared are considered matches. Valid values are: • *`MatchBlanks`* • *`NullNotAllowed`* (default)
`fieldName`	string	Required. Indicates which field to compare when determining if a record is similar enough to an existing record to be considered a match.

Field Name	Field Type	Description
matchingMethod	MatchingMethod (enumeration of type string)	Required. Defines how the fields are compared. Choose between the exact matching method and various fuzzy matching methods. Valid values are: • *Exact* • *FirstName* • *LastName* • *CompanyName* • *Phone* • *City* • *Street* • *Zip* • *Title* For details on each matching method, see "Matching Methods Used with Matching Rules" in the Salesforce Help.

Declarative Metadata Sample Definition

The following is a sample XML definition of a matching rule. A matching rule can be associated with either a standard or a custom object.

```
<?xml version="1.0" encoding="UTF-8"?>
<MatchingRules xmlns="http://soap.sforce.com/2006/04/metadata">
<matchingRules>
<fullName>AccountMatchingRule</fullName>
<label>Matching rule for accounts</label>
<description>this is sample rule description</description>
<matchingRuleItems>
<blankValueBehavior>NullNotAllowed</blankValueBehavior>
<fieldName>BillingCity</fieldName>
<matchingMethod>City</matchingMethod>
</matchingRuleItems>

<matchingRuleItems>
<blankValueBehavior>NullNotAllowed</blankValueBehavior>
<fieldName>Name</fieldName>
<matchingMethod>CompanyName</matchingMethod>
</matchingRuleItems>

<ruleStatus>Inactive</ruleStatus>
</matchingRules>
</MatchingRules>
```

The following `package.xml` shows how to reference a matching rule by name. It specifies the type name of MatchingRule.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
<types>
<members>Account.AccountMatchingRule</members>
<name>MatchingRule</name>
```

```
</types>
<version>64.0</version>
</Package>
```

The following `package.xml` shows how to reference all matching rules by specifying the plural MatchingRules type name and using a wildcard to include all members.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
<types>
<members>*</members>
<name>MatchingRules</name>
</types>
<version>64.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

MessagingChannel

Represents the metadata associated with an Embedded Service Messaging channel.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

MessagingChannel components have the suffix `messagingChannel` and are stored in the `messagingChannels` folder.

Version

MessagingChannel components are available in API version 55.0 and later.

Special Access Rules

This type is available if your org has the "Configure Messaging" and "View Setup and Configuration" permissions for Messaging enabled.

Fields

Field Name	Description
`automatedResponses`	**Field Type** MessagingAutoResponse[] **Description** The auto-responses associated with the messaging channel.
`customParameters`	**Field Type** MessagingChannelCustomParameter[] **Description** The custom parameters associated with the messaging channel.
`description`	**Field Type** string **Description** The channel description.
`embeddedConfig`	**Field Type** EmbeddedConfig[] **Description** The settings associated with the messaging channel.
`masterLabel`	**Field Type** string **Description** Required. The channel label.
`messagingChannelType`	**Field Type** MessagingChannelType (enumeration of type string) **Description** Required. Values are: • `Custom`—Bring Your Own Channel for Messaging or Bring Your Own Channel for CCaaS. Available in API version 61.0 and later. • `EmbeddedMessaging` • `Voice`—Service Cloud Voice. Available in API version 58.0 and later. Third-party Messaging channels in Salesforce, such as WhatsApp and Facebook Messenger, don't use this metadata type.
`messagingKeywords`	**Field Type** MessagingKeyword[]

Field Name	Description
	Description Keywords associated with the messaging channel.
`sessionHandlerFlow`	**Field Type** string **Description** The Omni-Channel flow used to route the channel's messaging sessions.
`sessionHandlerQueue`	**Field Type** string **Description** Required. The queue used to route messages. If a sessionHandlerFlow is also selected, sessionHandlerQueue is the fallback queue used if a message can't be routed using the selected flow.
`sessionHandlerType`	**Field Type** MessagingSessionHandlerType (enumeration of type string) **Description** Required. The method used to route messages in the channel. Values are: • `Flow` • `Queue`
`standardParameters`	**Field Type** MessagingChannelStandardParameter[] **Description** Parameters added to the messaging channel.

EmbeddedConfig

Represents settings specific to an embedded messaging channel.

Field Name	Description
`isEstimatedWaitTimeEnabled`	**Field Type** Boolean **Description** Indicates whether estimated wait time is enabled. The default value is false.
`authMode`	**Field Type** EmbeddedServiceAuthModeType (enumeration of type string)

Field Name	Description
	Description Required. Auth/UnAuth Mode Types For Embedded Chat. Valid values are: • Auth • UnAuth
`verifiedUserJwtExpirationTime`	**Field Type** Integer **Description** If `authMode` is set to `true`, this value represents the JWT expiry value in minutes.
`anonymousUserJwtExpirationTime`	**Field Type** Integer **Description** If `authMode` is set to `false`, this value represents the JWT expiry value in minutes.
`isAttachmentUploadEnabled`	**Field Type** Boolean **Description** Indicates whether inbound file attachments are allowed. The default value is false.
`isSaveTranscriptEnabled`	**Field Type** Boolean **Description** Indicates whether transcripts can be downloaded. The default value is false.
`messagingAuthorizations`	**Field Type** MessagingAuthorization **Description** Represents authorization settings specific to a given MessagingChannel.

MessagingAuthorization

This junction entity stores the messaging channel authorization for a Messaging Channel (Embedded Messaging Channel). On this entity, we configure different authorization methods supported by the Messaging for In-App and Web `(Embedded Messaging) channel. This entity is available in API version 62.0 or later.

Field Name	Description
authIdentifier	**Field Type** String **Description** Required. Represents an identifier for the MessagingAuthorization entity. The value must begin with a letter. Only alphanumeric characters and _ are allowed.
authProviderName	**Field Type** String **Description** This field should never be set. Although this field is exposed, it isn't supported. The field represents the name of the AuthProvider that's referenced in the MessagingAuthorization. It's a required field if authorizationType is set to AuthProvider.
authorizationType	**Field Type** MessagingAuthorizationType (enumeration of type string) **Description** Required. Indicates whether MessagingAuthorization has been configured to reference a PublicKeyCertificateSet or an AuthProvider. Valid values are: • `PublicKeyCertificateSet` - Indicates that a PublicKeyCertificateSet is referenced in the MessagingAuthorization entity. This is the only option that will lead to a fully functional User Verification setup. • `AuthProvider` - This option should never be selected. It indicates that an AuthProvider is referenced in the MessagingAuthorization, but we don't support it.
enabled	**Field Type** Boolean **Description** Indicates whether messaging authorization is enabled. Defaults to false.
publicKeyCertificateSetName	**Field Type** String **Description** Name of the public key certificate set that's referenced in the MessagingAuthorization entity. This is a required field if `authorizationType` is set to `PublicKeyCertificateSet`.

MessagingAutoResponse

Represents an automated response used in a channel.

Field Name	Description
autoResponseContentType	**Field Type** string **Description** The content type of the auto-response: TextResponse or MessageDefinition.
language	**Field Type** string **Description** The language of the auto response.
messageDefinitionName	**Field Type** string **Description** The name of the messaging component.
`response`	**Field Type** string **Description** The text of the auto response.
responseTimeoutInMins	**Field Type** integer **Description** The number of minutes after which a response can no longer be sent. The value can range from 5 to 60.
`type`	**Field Type** MessagingAutoResponseType (enumeration of type string) **Description** Required. The type of response, which determines when it's used in a messaging session. Values are: • `AgentEndEngagementResponse` • `AgentEngagedResponse` • `InitialResponse`

MessagingChannelCustomParameter

Represent a custom parameter added to a channel.

Field Name	Description
`actionParameterMappings`	**Field Type** MessagingChannelActionParameterMapping[] **Description** The mapping used to map the parameter value to a flow or task.
`externalParameterName`	**Field Type** string **Description** Required. The external name of the parameter.
`masterLabel`	**Field Type** string **Description** Required. The label of the parameter.
`maxLength`	**Field Type** int **Description** The maximum length of the parameter value.
`name`	**Field Type** string **Description** Required. The name of the parameter.
`parameterDataType`	**Field Type** FlowDataType (enumeration of type string) **Description** Required. The format of the parameter. Values are: • `Apex` • `Boolean` • `Currency` • `Date` • `DateTime` • `Multipicklist` • `Number` • `Picklist` • `SObject` • `String`

MessagingChannelActionParameterMapping

Represents a mapping between a parameter and an Omni-Channel flow or agent task.

Field Name	Description
`actionParameterName`	**Field Type** string **Description** Required. The name of the flow that the custom or standard parameters are mapped to.

MessagingChannelStandardParameter

Represents a standard parameter used to pass information into a channel.

Field Name	Description
`actionParameterMappings`	**Field Type** MessagingChannelActionParameterMapping[] **Description** The mapping associated with the parameter.
`externalInteractionId`	**Field Type** MessagingChannelStandardParameterType (enumeration of type string) **Description** An ID assigned to the external interaction, such as a campaign ID.
`externalInteractionName`	**Field Type** MessagingChannelStandardParameterType (enumeration of type string) **Description** The name of the external interaction, such as a campaign name.
`externalInteractionType`	**Field Type** MessagingChannelStandardParameterType (enumeration of type string) **Description** The type of external interaction, such as MarketingCampaign.
`parameterType`	**Field Type** MessagingChannelStandardParameterType (enumeration of type string) **Description** Required. The type of parameter. Values are: • `Email` • `FirstName`

Field Name	Description
	• LastName • Subject

MessagingKeyword

Represents settings specific to an EmbeddedMessaging MessagingChannel. Available in API version 62 or later.

Field Name	Description
keywordType	**Field Type** MessagingKeywordType (enumeration of type string) **Description** Required. The type of messaging keyword for the auto response. Valid values are: • OptIn • DoubleOptIn • OptOut • Help • Custom
language	**Field Type** String **Description** Required. The language of the messaging keyword
keyword	**Field Type** String **Description** The messaging keyword value.

Declarative Metadata Sample Definition

The following is an example of a MessagingChannel component. This messaging channel passes custom and standard parameters from the messaging channel to a flow, and it routes to a flow with a fallback queue.

```
<?xml version="1.0" encoding="UTF-8"?>
<MessagingChannel xmlns="http://soap.sforce.com/2006/04/metadata">
    <description>Test in-app messaging channel</description>
    <masterLabel>TestInAppChannel</masterLabel>
    <messagingChannelType>EmbeddedMessaging</messagingChannelType>
    <sessionHandlerQueue>Demo_Queue</sessionHandlerQueue>
    <sessionHandlerType>Queue</sessionHandlerType>
    <embeddedConfig>
```

```
        <authMode>Auth</authMode>
        <isAttachmentUploadEnabled>true</isAttachmentUploadEnabled>
        <isSaveTranscriptEnabled>false</isSaveTranscriptEnabled>
        <isEstimatedWaitTimeEnabled>false</isEstimatedWaitTimeEnabled>
        <verifiedUserJwtExpirationTime>360</verifiedUserJwtExpirationTime>
        <messagingAuthorizations>
            <authorizationType>PublicKeyCertificateSet</authorizationType>
            <authProviderName></authProviderName>
            <publicKeyCertificateSetName>pcks1</publicKeyCertificateSetName>
            <enabled>false</enabled>
            <authIdentifier>auth_identifier_one</authIdentifier>
        </messagingAuthorizations>
    </embeddedConfig>
    <automatedResponses>
        <autoResponseContentType>MessageDefinition</autoResponseContentType>
        <messageDefinitionName>Sample</messageDefinitionName>
        <type>EndUserInactiveResponse</type>
        <responseTimeoutInMins>10</responseTimeoutInMins>
    </automatedResponses>
    <automatedResponses>
        <autoResponseContentType>MessageDefinition</autoResponseContentType>
        <messageDefinitionName>Sample</messageDefinitionName>
        <type>InitialResponse</type>
    </automatedResponses>
    <automatedResponses>
        <autoResponseContentType>MessageDefinition</autoResponseContentType>
        <messageDefinitionName>Sample</messageDefinitionName>
        <type>AgentEndEngagementResponse</type>
    </automatedResponses>
    <automatedResponses>
        <autoResponseContentType>MessageDefinition</autoResponseContentType>
        <messageDefinitionName>Sample</messageDefinitionName>
        <type>AgentEngagedResponse</type>
    </automatedResponses>
    <automatedResponses>
        <autoResponseContentType>TextResponse</autoResponseContentType>
        <language>en_US</language>
        <response>You've opted out of receiving messages from us, so we won't contact you
again.</response>
        <type>OptOutConfirmation</type>
    </automatedResponses>
    <automatedResponses>
        <autoResponseContentType>TextResponse</autoResponseContentType>
        <language>en_US</language>
        <response>Custom response1</response>
        <type>CustomResponse</type>
    </automatedResponses>
    <automatedResponses>
        <autoResponseContentType>TextResponse</autoResponseContentType>
        <language>en_US</language>
        <response>Opt In Confirmation response</response>
        <type>OptInConfirmation</type>
    </automatedResponses>
    <automatedResponses>
```

```
        <autoResponseContentType>TextResponse</autoResponseContentType>
        <language>en_US</language>
        <response>Text STOP to opt out of further messages.</response>
        <type>HelpResponse</type>
    </automatedResponses>
    <messagingKeywords>
        <keyword>stopall</keyword>
        <keyword>cancel</keyword>
        <keyword>stop</keyword>
        <keyword>unsubscribe</keyword>
        <keyword>end</keyword>
        <keyword>quit</keyword>
        <keywordType>OptOut</keywordType>
        <language>en_US</language>
    </messagingKeywords>
    <messagingKeywords>
        <keyword>help</keyword>
        <keywordType>Help</keywordType>
        <language>en_US</language>
    </messagingKeywords>
    <messagingKeywords>
        <keyword>customkeyword1</keyword>
        <keywordType>Custom</keywordType>
        <language>en_US</language>
    </messagingKeywords>
    <messagingKeywords>
        <keyword>OptInkeyword1</keyword>
        <keywordType>OptIn</keywordType>
        <language>en_US</language>
    </messagingKeywords>
</MessagingChannel>
```

If you route the messaging channel to a queue, there's no fallback flow.

```
<?xml version="1.0" encoding="UTF-8"?>
<MessagingChannel xmlns="http://soap.sforce.com/2006/04/metadata">
    <masterLabel>EmbeddedChannel2</masterLabel>
    <messagingChannelType>EmbeddedMessaging</messagingChannelType>
    <sessionHandlerQueue>DemoQueueName</sessionHandlerQueue>
    <sessionHandlerType>Queue</sessionHandlerType>
</MessagingChannel>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>MessagingChannel</name>
    </types>
    <version>55.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

Metadata

The base class for all metadata types. You can't edit this object. A component is an instance of a metadata type.

Metadata is analogous to sObject, which represents all standard objects. Metadata represents all components and fields in the Metadata API. Instead of identifying each component with an ID, each custom object or custom field has a unique `fullName`, which must be distinct from standard object names, as it must be when you create custom objects or custom fields in the Salesforce user interface.

Version

Metadata components are available in API version 10.0 and later.

Fields

Field Name	Field Type	Description
`fullName`	string	Required. The name of the component. For components with parent objects, such as fields and list views, the name must specify the name of the parent, for example `Account.FirstName`. The `__c` suffix must be appended to custom object names and custom field names when you're setting the `fullName`. For example, a custom field in a custom object could have a `fullName` of `MyCustomObject__c.MyCustomField__c`. To reference a component in a package, prepend the package's namespace prefix to the component name in the `fullName` field. Use the following syntax: ***`namespacePrefix__ComponentName`***. For example, for the custom field component `MyCustomObject__c.MyCustomField__c` and the namespace `MyNS`, the full name is `MyNS__MyCustomObject__c.MyCustomField__c`. A namespace prefix is a 1-character to 15-character alphanumeric identifier that distinguishes your package and its contents from other publishers' packages. For more information, see Create and Register Your Namespace for Second-Generation Managed Packages.

Wildcard Support in the Manifest File

This metadata type doesn't support the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SEE ALSO:

- CustomObject
- CustomField
- MetadataWithContent

MetadataWithContent

MetadataWithContent is the base type for all metadata types that contain content, such as documents or email templates. It extends Metadata. You can't edit this object.

Version

MetadataWithContent components are available in API version 14.0 and later.

Fields

Field Name	Field Type	Description
`content`	base64Binary	Base 64-encoded binary data. Before making an API call, client applications must encode the binary attachment data as base64. Upon receiving a response, client applications must decode the base64 data to binary. This conversion is handled for you by a SOAP client.
`fullName`	string	Required. The name of the component. The `fullName` can contain only underscores and alphanumeric characters. It must be unique, begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. Inherited from the Metadata component, this field isn't defined in the WSDL for this component. It must be specified when creating, updating, or deleting. See create() to see an example of this field specified for a call.

Wildcard Support in the Manifest File

This metadata type doesn't support the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SEE ALSO:

- Metadata

MfgProgramTemplate

Represents a definition of a program to create a program-based business. A program-based business, also known as a Manufacturing Program, enables manufacturers to drive their business models with forecasting tools and manage the end-to-end sales process efficiently.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

MfgProgramTemplate components have the suffix `.mfgProgramTemplate` and are stored in the `MfgProgramTemplate` folder.

Version

MfgProgramTemplate components are available in API version 54.0 and later.

Special Access Rules

The program-based business feature setting for Manufacturing Cloud is required to create a program template.

Fields

Field Name	Description
`description`	**Field Type** string **Description** The description of the manufacturing program template.
`programTemplateItems`	**Field Type** MfgProgramTemplateItem[] **Description** The list of templates associated with the manufacturing program template.
`programTemplateName`	**Field Type** string **Description** Required. The unique identifier for the manufacturing program template.
`status`	**Field Type** MfgProgramTemplateStatus (enumeration of type string)

Field Name	Description
	Description Required. The status of the manufacturing program template. Values are: • `Active` • `Draft` • `Inactive` The default value is `Active`.

MfgProgramTemplateItem

A program template item defines each of the templates associated with a manufacturing program. A template item includes program details, such as a data transformation type and a display order. Transformation type is the method to forecast business visibility to manufacturers.

Field Name	Description
`advAccountForecastSet`	**Field Type** string **Description** The forecast set associated with the transformation.
`description`	**Field Type** string **Description** The description of the manufacturing program template item.
`templateItemName`	**Field Type** string **Description** Required. The name of the manufacturing program template item.
`transformationDisplayOrder`	**Field Type** int **Description** Required. The display order of the transformation in the manufacturing program template.

Field Name	Description
`transformationType`	**Field Type** MfgProgramTransformationType (enumeration of type string) **Description** Required. Specifies the type of transformation. Values are: • `BusinessTransformation` • `ForecastSetRelation`

Declarative Metadata Sample Definition

The following is an example of a MfgProgramTemplate component.

```
<?xml version="1.0" encoding="UTF-8"?>
<MfgProgramTemplate xmlns="http://soap.sforce.com/2006/04/metadata"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
     <description>Program Template</description>
     <programTemplateItems>
         <templateItemName>Template Item #1</templateItemName>
         <transformationDisplayOrder>1</transformationDisplayOrder>
         <transformationType>BusinessTransformation</transformationType>
         <description>Program Template Item</description>
     </programTemplateItems>
     <programTemplateName>Sample Program Template</programTemplateName>
     <status>Draft</status>
</MfgProgramTemplate>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
     <types>
         <members>*</members>
         <name>MfgProgramTemplate</name>
     </types>
     <version>54.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

MilestoneType

Represents the name and description of a milestone, which you can use in an entitlement process to track important steps in cases.

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

Milestone types are stored in the `milestoneTypes` directory of the corresponding package directory. The extension is `.milestoneType`.

Version

MilestoneType on page 1441 is available in API version 27.0 and later.

Fields

Field Name	Field Type	Description
`description`	string	The description of the milestone.
`RecurrenceType`	MilestoneRecurrenceType (enumeration of type string)	The type of recurrence for the milestone. Available in API version 29.0 and later. Valid values are: • `none`—Specifies no recurrence for the milestone. The milestone occurs only one time until the entitlement process exits. • `recursIndependently`—Specifies independent recurrence for the milestone. • `recursChained`—Specifies sequential recurrence for the milestone.

Declarative Metadata Sample Definition

Here's a sample milestone type.

```
<?xml version="1.0" encoding="UTF-8"?>
<MilestoneType xmlns="http://soap.sforce.com/2006/04/metadata">
    <description>First Response Time</description>
</MilestoneType>
```

And, here's the sample `package.xml` file that references the MilestoneType component definition:

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>* or a valid name of a milestone type</members>
        <name>MilestoneType</name>
    </types>
    <version>29.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

MlDomain

Represents an Einstein Intent Set. This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

MlDomain components have the suffix `.mlDomain` and are stored in the `mlDomains` folder.

Version

MlDomain components are available in API version 43.0 and later.

Special Access Rules

This object is available only if Chat and Einstein Bots are enabled in your org.

Fields

Field Name	Field Type	Description
`description`	string	Einstein Intent Set description.
`label`	string	Einstein Intent Set name.
`mlIntents`	MlIntent[]	List of intents under this Einstein Intent Set.
`mlSlotClasses`	MlSlotClass[]	List of entities under this Einstein Intent Set.

MlIntent

An intent in an Einstein Intent Set.

Field Name	Field Type	Description
`description`	string	Einstein Intent Set description.
`developerName`	string	Required. This unique name prevents conflicts with other Einstein Intent Sets associated with the same bot version. This name can contain only underscores and alphanumeric characters and must be unique in your org. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.

Field Name	Field Type	Description
		Note: Only users with View DeveloperName OR View Setup and Configuration permission can view, group, sort, and filter this field.
`label`	string	Einstein Intent Set name.
`mlIntentUtterances`	MlIntentUtterance[]	List of customer inputs for this intent.
`relatedMlIntents`	MlRelatedIntent[]	List of intents within an Einstein Intent Set used to expand customer inputs for this intent. Only intents within local Einstein Intent Sets have related intents.

MlIntentUtterance

A customer input for this intent.

Field Name	Field Type	Description
`utterance`	string	A customer input or natural language query that triggers the parent intent.

MlRelatedIntent

An intent in an Einstein Intent Set used to expand customer inputs for this intent. Only intents within local Einstein Intent Sets have related intents.

Field Name	Field Type	Description
`relatedMlIntent`	string	Name of the intent that is used to extend the customer inputs of the current parent intent.

MlSlotClass

An entity in this Einstein Intent Set.

Field Name	Field Type	Description
`dataType`	MlSlotClassDataType (enumeration of type string)	A list of the data types available for the MlSlotClass. Valid values are: • `Text` • `Number` • `Boolean` • `Date` • `DateTime` • `Currency`
`description`	string	A description of an Einstein Bot entity.

Field Name	Field Type	Description
`developerName`	string	Required. This unique name prevents conflicts with other entities in an Einstein Intent Set. This name can contain only underscores and alphanumeric characters, and must be unique in your org. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. Note: Only users with View DeveloperName OR View Setup and Configuration permission can view, group, sort, and filter this field.
`extractionRegex`	string	Regular expression used to extract an entity when the type is set to `Pattern`.
`extractionType`	MlSlotClassExtractionType (enumeration of type string)	Required. Valid values are: • `Pattern` • `Value`
`label`	string	Label that identifies an entity throughout the Salesforce user interface.
`mlSlotClassValues`	MlSlotClassValue[]	List of entity values associated with an entity of type `Value`.

MlSlotClassValue

An entity value associated with an entity of type `Value`.

Field Name	Field Type	Description
`synonymGroup`	SynonymGroup	Represents a list of terms or synonyms for the current entity value.
`value`	string	Single value used to extract an entity of type `Value`.

SynonymGroup

Represents a group of synonymous words or phrases.

Field Name	Field Type	Description
`languages`	Language (enumeration of type string)	Required. Specifies the languages the value list applies to. If value list items are specific to a single language, specify only that language. If the value list items apply to multiple languages, specify multiple languages for one value list.
`terms`	string	Required. A word or phrase synonymous with other terms in the value list.

Declarative Metadata Sample Definition

The following is an example of an MlDomain.

```
<?xml version="1.0" encoding="UTF-8"?>
<MlDomain xmlns="http://soap.sforce.com/2006/04/metadata">
```

```
    <label>TestDomainMetadata</label>
    <description>This is domain 2 for metadata testing</description>
    <mlIntents>
        <developerName>Test_Intent_New</developerName>
        <label>Test Intent New</label>
        <mlIntentUtterances>
            <utterance>Utterance Hello</utterance>
        </mlIntentUtterances>
        <mlIntentUtterances>
            <utterance>Utterance Hi</utterance>
        </mlIntentUtterances>
        <mlIntentUtterances>
            <utterance>Utterance What</utterance>
        </mlIntentUtterances>
    </mlIntents>
    <mlIntents>
        <developerName>Test_Intent_New2</developerName>
        <label>Test Intent New 2</label>
    </mlIntents>
    <mlSlotClasses>
        <developerName>Test_Entity1</developerName>
        <label>Test Entity 1</label>
        <extractionType>Value</extractionType>
        <mlSlotClassValues>
                <value>Choice value 1</value>
            </mlSlotClassValues>
            <mlSlotClassValues>
                <value>Choice value 2</value>
            </mlSlotClassValues>
    </mlSlotClasses>
    <mlSlotClasses>
        <developerName>Test_Entity2</developerName>
        <label>Test Entity 2</label>
        <extractionType>Pattern</extractionType>
    </mlSlotClasses>
    <mlSlotClasses>
            <dataType>Text</dataType>
            <description>Valid Email Address</description>
            <developerName>Email</developerName>
            <extractionRegex>\b[A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-Z]{2,}\b</extractionRegex>
            <extractionType>Pattern</extractionType>
            <label>Email</label>
    </mlSlotClasses>
    <mlSlotClasses>
        <developerName>airport</developerName>
        <extractionType>Value</extractionType>
        <label>airport</label>
        <mlSlotClassValues>
            <synonymGroup>
                <languages>en_US</languages>
                <terms>San Francisco</terms>
                <terms>The City</terms>
            </synonymGroup>
            <value>SFO</value>
```

```
        </mlSlotClassValues>
        <mlSlotClassValues>
            <synonymGroup>
                <languages>en_US</languages>
                <terms>Oakland</terms>
                <terms>The Town</terms>
            </synonymGroup>
            <value>OAK</value>
        </mlSlotClassValues>
    </mlSlotClasses>
</MlDomain>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>TestDomainMetadata</members>
        <name>MlDomain</name>
    </types>
    <version>43.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

MLDataDefinition

Represents a modeling data definition, which specifies the data used to create a model. Such data can include filters, fields to include, fields to exclude, and so on. This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

MLDataDefinition components have the suffix `.mlDataDefinition` and are stored in the `mlDataDefinitions` folder.

Version

MLDataDefinition is available in API version 50.0 and later.

Fields

Field Name	Field Type	Description
`developerName`	string	Required. Represents the name of the data definition. Can contain only underscores and alphanumeric characters and must be unique in your org. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.

Field Name	Field Type	Description
		Note: Only users with View DeveloperName OR View Setup and Configuration permission can view, group, sort, and filter this field.
`entityDeveloperName`	string	Required. The developer name of the object from which the model data is retrieved. Note: After the MLDataDefinition entity is created, `entityDeveloperName` can't be updated.
`excludedFields`	string[]	Fields that are excluded from the model.
`includedFields`	string[]	Fields that are included in the model.
`joinFields`	MLField[]	Reserved for future use.
`parentDefinitionDevName`	string	Reserved for future use.
`scoringFilter`	MLFilter	Specifies records to which the prediction scores are written.
`segmentFilter`	MLFilter	This field further filters data used in training and scoring when `segmentFilter` is combined with both `scoringFilter` and `trainingFilter`. For example, select all records in a specific region.
`trainingFilter`	MLFilter	Specifies the records that make up the training set.
`type`	MLDataDefinitionType (enumeration of type string)	Required. Valid values are: • `Candidate` • `Interaction` • `Prediction` • `Recipient` Note: After the model is created, `type` can't be updated.

MLField

Represents a single field in the data definition. Available in API version 50.0 and later.

Field Name	Field Type	Description
`entity`	string	Required. The object that contains the field.
`field`	string	Required. The name of the field.
`relatedField`	MLField	Reserved for future use.
`relationType`	MLRelationType (enumeration of type string)	Reserved for future use. Valid values are: • `Full` • `Inner`

Field Name	Field Type	Description
		• `Leftinner` • `Leftouter`
`type`	MLFieldType (enumeration of type string)	Required. How the field is used in a prediction. Valid values are: • `Excluded` • `Expression` • `Included` • `Join` • `Prediction` • `Pushback` • `Related` • `SourceDate`

MLFilter

Represents a data filter based on a data comparison. For each comparison, there's a left-hand element, an operator, and a right-hand element. For each record, only one of these left-hand elements is populated: `lhFilter`, `lhPredictionField`, or `lhValue`. Similarly, for each record, only one of these right-hand elements is populated: `rhFilter`, `rhPredictionField`, or `rhValue`. Available in API version 50.0 and later.

Field Name	Field Type	Description
`filterName`	string	Required. Name of the filter.
`lhFilter`	MLFilter	Left-hand filter condition.
`lhPredictionField`	string	Left-hand prediction field.
`lhType`	AIValueType (enumeration of type string)	The value type if a left-hand value is specified. Valid values are: • `Boolean` • `Comparison` • `Currency` • `Date` • `DateTime` • `Number` • `String` • `Supplier` • `Varchar`
`lhUnit`	AIFilterUnit (enumeration of type string)	The unit if a left-hand filter is specified. Valid values are: • `Milliseconds` • `Seconds`

Field Name	Field Type	Description
		• Minutes • Hours • Days • Weeks • Months • Years
lhValue	string	The left-hand value.
operation	AIFilterOperation (enumeration of type string)	Required. Valid values are: • And • Or • Not • LessThan • LessThanOrEqual • GreaterThan • GreaterThanOrEqual • Equals • NotEquals • Add • Subtract • Multiply • Divide • IsNull • IsNotNull • StartsWith • EndsWith • Contains • Concat • DoesNotContain • Between • In
rhFilter	MLFilter	Right-hand filter condition.
rhPredictionField	string	Right-hand prediction field.
rhType	AIValueType (enumeration of type string)	The value type if a right-hand value is specified. Valid values are: • Boolean • Comparison • Currency

Field Name	Field Type	Description
		• `Date` • `DateTime` • `Number` • `String` • `Supplier` • `Varchar`
`rhUnit`	AlFilterUnit (enumeration of type string)	The unit if a right-hand filter is specified. Valid values are: • `Milliseconds` • `Seconds` • `Minutes` • `Hours` • `Days` • `Weeks` • `Months` • `Years`
`rhValue`	string	The right-hand value.
`sortOrder`	int	Specifies the order of operations for evaluating the expressions. For example, if you have two conditions, this field specifies which condition is evaluated first.

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

MLPredictionDefinition

Represents a prediction definition that specifies details about the prediction. This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

MLPredictionDefinition components have the suffix `.mlPrediction` and are stored in the `mlPredictions` folder.

Version

MLPredictionDefinition is available in API version 50.0 and later.

Fields

Field Name	Field Type	Description
aiApplicationDeveloperName	string	Required. Represents the developer name of the parent AI application. Can contain only underscores and alphanumeric characters and must be unique in your org. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
description	string	Description of the prediction.
developerName	string	Required. Represents the name of the prediction definition. Can contain only underscores and alphanumeric characters and must be unique in your org. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. Note: Only users with View DeveloperName OR View Setup and Configuration permission can view, group, sort, and filter this field.
masterLabel	string	Label that identifies the ML prediction definition throughout the Salesforce user interface.
negativeExpression	MLFilter	Reserved for future use.
positiveExpression	MLFilter	Reserved for future use.
predictionField	string	Field that the prediction is based on.
priority	int	Reflects the priority of the MLPD object when an AIApplication has multiple child MLPDs. Nillable.
pushbackField	string	Field that the prediction writes scores to.
status	MLPredictionDefinitionStatus (enumeration of type string)	Required. The status of the prediction. Valid values are: • Enabled • Disabled • Draft
type	AIPredictionType (enumeration of type string)	Required. The type of model that returns the prediction values. Valid values are: • BinaryClassification • DeepLearningIntentClassification • DeepLearningNameEntityRecognition • GlobalDeepLearningIntentClassification • GlobalDeepLearningNameEntityRecognition • LanguageDetection • MulticlassClassification • Regression • ScoringSpecificOutcome

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

MobileApplicationDetail

Represents the packaging attributes for a mobile connected app. This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

MobileApplicationDetail components have the suffix `MobileApplicationDetail` and are stored in the `MobileApplicationDetails` folder.

Version

MobileApplicationDetail components are available in API version 47.0 and later.

Fields

Field Name	Field Type	Description
`applicationBinaryFile`	base64	Base 64-encoded binary data file for the mobile app.
`applicationBinaryFileName`	string	Filename for the mobile app binary data file.
`applicationBundleIdentifier`	string	iOS apps only: the unique application bundle identifier.
`applicationFileLength`	int	The length of the mobile app binary data file.
`applicationIconFile`	string	iOS apps only: the application icon.
`applicationIconFileName`	string	iOS apps only: the application icon filename.
`applicationInstallUrl`	string	URL to install the mobile app.
`devicePlatform`	DevicePlatformType (enumeration of type string)	Required. Platform that supports the mobile app. The valid values are: • `android` • `ios`
`deviceType`	string	Supported device type for mobile app. The valid values are: • `minitablet` • `phone` • `tablet`
`minimumOsVersion`	string	Minimum OS version required to install the mobile app.
`privateApp`	boolean	Specifies whether the mobile app is private (`true`) or not (`false`).

Field Name	Field Type	Description
version	string	Required. Version number of the mobile app.

Usage

When you create a connected app in Salesforce Classic or Lightning Experience and enter mobile app settings, those settings are stored in a MobileApplicationDetail component. In this example, the metadata retrieved for a connected app includes MobileApplicationDetail metadata.

```
<?xml version="1.0" encoding="UTF-8"?>
<<ConnectedApp xmlns="http://soap.sforce.com/2006/04/metadata">
  <contactEmail>paul.chen@salesforce.com</contactEmail>
  <label>MobileApplicationDetailConnectedApp</label>
  <mobileAppConfig>
    <applicationBinaryFile></applicationBinaryFile>
    <applicationInstallUrl>https://appstore.apple.com/MobileApplicationDetail
        </applicationInstallUrl>
    <devicePlatform>ios</devicePlatform>
    <deviceType>phone</deviceType>
    <privateApp>false</privateApp>
    <version>0.0.0.0</version>
  </mobileAppConfig>
< . mobileStartUrl>https://www.salesforce.com</mobileStartUrl>
</ConnectedApp>
```

Wildcard Support in the Manifest File

This metadata type doesn't support the wildcard character * (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

MobileSecurityAssignment

Represents the assignment of mobile app security policies to a profile. The policies apply to the Salesforce mobile app with Enhanced Mobile App Security enabled.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

MobileSecurityAssignment components have the suffix `.mobileSecurityAssignment` and are stored in the `mobileSecurityAssignments` folder.

Version

MobileSecurityAssignment components are available in API version 54.0 and later.

Special Access Rules

The Enhanced Mobile App Security add-on subscription and the Enforce Enhanced Mobile App Security user permission are required to use this metadata type.

Fields

Field Name	Description
`connectedApplication`	**Field Type** string **Description** The name of the connected app that's associated with the mobile security policies assignment.
`isProtected`	**Field Type** boolean **Description** An auto-generated value that doesn't impact the behavior of the metadata type. The default is `false`.
`masterLabel`	**Field Type** string **Description** Required. A user-friendly name for MobileSecurityAssignment, which is defined when the MobileSecurityAssignment component is created.
`profile`	**Field Type** string **Description** The profile that the mobile security policies are assigned to.

Declarative Metadata Sample Definition

The following is an example of a MobileSecurityAssignment component.

```
<?xml version="1.0" encoding="UTF-8"?>
<MobileSecurityAssignment xmlns="http://soap.sforce.com/2006/04/metadata">
    <connectedApplication>MyMobileConnectedApp</connectedApplication>
    <isProtected>false</isProtected>
    <masterLabel>MyMobileSecurityAssignment</masterLabel>
```

```
    <profile>admin</profile>
</MobileSecurityAssignment>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>MobileSecurityAssignment</name>
    </types>
    <version>61.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

MobileSecurityPolicy

Represents a mobile app security policy on the Salesforce mobile app with Enhanced Mobile App Security enabled. For a full description of each policy, see Enable and Configure Mobile App Security Policies.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

MobileSecurityPolicy components have the suffix `.mobileSecurityPolicy` and are stored in the `mobileSecurityPolicies` folder.

Version

MobileSecurityPolicy components are available in API version 53.0 and later.

Special Access Rules

The Enhanced Mobile App Security add-on subscription and the Enforce Enhanced Mobile App Security user permission are required to use this metadata type.

Fields

Field Name	Description
`effectiveDate`	**Field Type** dateTime **Description** The date that a mobile security policy is enforced.
`isEnabled`	**Field Type** boolean **Description** Required. Indicates whether the mobile security policy is enabled. The default value is `false`, which means that the policy is disabled.
`isProtected`	**Field Type** boolean **Description** An auto-generated value that doesn't impact the behavior of the metadata type. The default value is `false`.
`masterLabel`	**Field Type** string **Description** Required. A user-friendly name for MobileSecurityPolicy, which is defined when the MobileSecurityPolicy component is created.
`mobilePlatform`	**Field Type** MobileSecurityMobilePlatform (enumeration of type string) **Description** The mobile operating system of the mobile security policy. Values are: • `Android` • `iOS`
`mobileSecurityAssignment`	**Field Type** string **Description** The name of the mobile security assignment associated with the mobile security policy. See MobileSecurityAssignment on page 1454.
`ruleValue`	**Field Type** string

Field Name	Description
	Description Required. The value of the mobile security policy rule.
`ruleValueType`	**Field Type** MobileSecurityPolicyRuleValueType (enumeration of type string) **Description** Required. The type of mobile security policy rule. Values are: • `Boolean` • `Text` • `TextList`
`severityLevel`	**Field Type** MobileSecurityPolicySeverityLevel (enumeration of type string) **Description** Required. The severity level of a mobile security policy. Values are: • `Critical`— Wipes app data and logs user out • `Error`—Blocks access to the app until the issue is resolved, but doesn't log user out • `Info`— Blocks prohibited action or logs user action and informs user • `Warn`—Notifies the user of the violation and recommends how to resolve, but user is able to continue using the app
`type`	**Field Type** MobileSecurityPolicyType (enumeration of type string) **Description** Required. The type of mobile security policy. Values are: • `AllowedDeviceList` • `Block3dTouch` • `BlockCalendar` • `BlockCamera` • `BlockContacts` • `BlockCustomKeyboard` • `BlockFileBackup` • `BlockMicrophone` • `BlockOsSharing`

Field Name	Description
	• BlockedDeviceList • BrowserUriScheme • CheckBiometric • DevicePasscode • DisableUrlCaching • JailbrokenDevice • LogCertPin • LogEmail • LogPhonecall • LogPolicyResult • LogScreenshot • LogTextmessage • LogoutAfterRestart • LogoutOnBiometricChange • MalwareDetection • ManInMiddle • MaxOffline • MaximumAppVersion • MaximumOsVersion • MinimumAppVersion • MinimumOsVersion • MinimumSecurityPatchVersion • MininumAppVersion • PhonecallUriScheme • Screenshot

Declarative Metadata Sample Definition

The following is an example of a MobileSecurityPolicy component.

```
<?xml version="1.0" encoding="UTF-8"?>
<MobileSecurityPolicy xmlns="http://soap.sforce.com/2006/04/metadata">
    <effectiveDate>2022-08-09T22:04:56.000Z</effectiveDate>
    <isEnabled>true</isEnabled>
    <isProtected>false</isProtected>
    <masterLabel>MyMobileSecurityPolicy</masterLabel>
    <mobileSecurityAssignment>MyMobileSecurityAssignment</mobileSecurityAssignment>
    <ruleValue>true</ruleValue>
    <ruleValueType>Boolean</ruleValueType>
    <severityLevel>info</severityLevel>
    <type>BlockCalendar</type>
```

```
    <mobilePlatform>Android</mobilePlatform>
</MobileSecurityPolicy>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>MobileSecurityPolicy</name>
    </types>
    <version>61.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

MobSecurityCertPinConfig

Represents the authentication server certificate pin configuration on the Salesforce mobile app with Enhanced Mobile Security.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

MobSecurityCertPinConfig components have the suffix `.mobSecurityCertPinConfig` and are stored in the `mobSecurityCertPinConfigs` folder.

Version

MobSecurityCertPinConfig components are available in API version 53.0 and later.

Special Access Rules

The Enhanced Mobile App Security add-on subscription and the Enforce Enhanced Mobile App Security user permission are required to use this metadata type.

Fields

Field Name	Description
`certificateHash`	**Field Type** string **Description** Required. The unique identifier for the certificate.
`domainName`	**Field Type** string **Description** Required. The name of the domain for the server that you want to pin the certificate to. For example, `https://MyDomainName.my.salesforce.com`.
`isEnabled`	**Field Type** boolean **Description** Required. Indicates whether authentication server certificate pinning is enabled. The default value is `false`, which means that certificate pinning is disabled.
`isProtected`	**Field Type** boolean **Description** An auto-generated value that doesn't impact the behavior of the metadata type. The default value is `false`.
`isSubdomainIncluded`	**Field Type** boolean **Description** Required. Indicates whether subdomains use the same certificate pinning configuration as the specified `domainName`. The default value is `false`.
`masterLabel`	**Field Type** string **Description** Required. A user-friendly name for MobSecurityCertPinConfig, which is defined when the MobSecurityCertPinConfig component is created.
`mobilePlatform`	**Field Type** MobileSecurityMobilePlatform (enumeration of type string)

Field Name	Description
	Description The mobile operating system. Values are: • `Android` • `iOS`
`mobileSecurityAssignment`	**Field Type** string **Description** The name of the mobile security assignment associated with the mobile security policy. See MobileSecurityAssignment on page 1454.
`severityLevel`	**Field Type** MobileSecurityPolicySeverityLevel (enumeration of type string) **Description** Required. The severity level of the mobile security policy. Values are: • `Critical`— Wipes app data and logs user out • `Error`—Blocks access to the app until the issue is resolved, but doesn't log user out • `Info`— Blocks prohibited action or logs user action and informs user • `Warn`—Notifies the user of the violation and recommends how to resolve, but user is able to continue using the app
`type`	**Field Type** MobileSecurityCertPinType (enumeration of type string) **Description** Required. The type of pin. Values are: • `AuthServer` • `Resource`

Declarative Metadata Sample Definition

The following is an example of a MobSecurityCertPinConfig component.

```
<?xml version="1.0" encoding="UTF-8"?>
<MobileSecurityCertPinConfig xmlns="http://soap.sforce.com/2006/04/metadata">
    <certificateHash>AaBbCcDdEeFfGg</certificateHash>
    <domainName>login.salesforce.com</domainName>
    <isEnabled>true</isEnabled>
```

```
    <isProtected>false</isProtected>
    <masterLabel>AuthenticationServerCertificatePinning</masterLabel>
    <mobilePlatform>Android</mobilePlatform>
    <mobileSecurityAssignment>MyMobileSecurityAssignment</mobileSecurityAssignment>
    <severityLevel>info</severityLevel>
    <type>AuthServer</type>
</MobileSecurityCertPinConfig>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>MobileSecurityCertPinConfig</name>
    </types>
    <version>61.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ModerationRule

Represents a rule used in your Experience Cloud site to moderate member-generated content. Each rule specifies the member-generated content the rule applies to, the criteria to enforce the rule on, and the moderation action to take. Moderation rules help protect your site from spammers, bots, and offensive or inappropriate content. This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Moderation rules created with the Metadata API are more powerful than moderation rules set up in the Experience Management UI. For example, in the UI you could create a rule that moderates posts and comments. In the Metadata API you could create a rule that moderates only the Link Name of a Link feed type. Use the Metadata API to express complex moderation rules.

Important: Don't update moderation rules you create using the Metadata API in the Experience Management UI. If you do, you overwrite relevant Metadata API fields or the fields are ignored.

Keep the following things in mind when creating moderation rules:

- Your org can have up to 30 rules. This limit is per org, not per site. This limit includes both content rules and rate rules.
- Each rule can have up to three keyword criteria.
- Rules that block content run first, followed by rules to review and approve content, then rules that replace content, and last by rules that flag content. If two or more rules perform the same action, the oldest rule runs first, based on the date the rule was created. Rules to replace content don't run when the content also applies to a review rule—we want community managers to review the original content.

File Suffix and Directory Location

ModerationRule components have the suffix `.rule` and are stored in the `moderation` directory of the corresponding package directory. The file name format follows *`site_name`*`.`*`moderation_rule_developer_name`*`.rule`.

Version

ModerationRule components are available in API version 36.0 and later.

Special Access Rules

To view, create, edit, and delete moderation rules, you need the Manage Experiences or Create and Set Up Experiences permission. As of Spring '20 and later, only users with permission to edit moderation rules can access this object.

Fields

Field Name	Field Type	Description
`action`	ModerationRuleAction (enumeration of type string)	Required. Indicates the moderation action that you want to take. The valid values are: • `Block` • `Review` • `Replace` • `Flag` • `FreezeAndNotify` (Reserved for future use.)
`actionLimit`	int	Indicates the moderation action limit. Available in API 39.0 and later.
`active`	boolean	Required. Indicates whether the moderation rule is active (`true`) or inactive (`false`).
`description`	string	A description of the moderation rule.
`entitiesAndFields`	ModerateEntityField[]	Indicates the types of user-generated content this moderation rule applies to.
`masterLabel`	string	Required. Label for the moderation rule.
`notifyLimit`	int	Indicates the notification limit of the moderation rule. Available in API 39.0 and later.
`userCriteria`	string	Represents the member criteria to use in moderation rules. Available in API 39.0 and later.
`userMessage`	string	The message you want your members to see when their content is blocked. Use the *`%BLOCKED_KEYWORD%`* variable to display up to five blocked words in the user message. If you don't specify a message, the member sees the standard message: "You can't use *`%BLOCKED_KEYWORD%`* or other inappropriate words in this site. Review your content and try again."

ModeratedEntityField

The fields and entities you want to moderate.

Field Name	Field Type	Description
`entityName`	string	Required. Indicates the types of user-generated content the moderation rule applies to. Post and comments only apply to content created in groups and user profiles. All feed types, such as polls and links, are supported.
`fieldName`	string	Indicates the field the moderation rule applies to. Note: To moderate feed posts, use `entityName FeedItem` with `fieldName RawBody`. To moderate feed comments, use `entityName FeedComment` with `fieldName RawCommentBody`. The `RawBody` and `RawCommentBody` fields aren't available in any other API.
`keywordList`	KeywordList string	Indicates the keyword list that you want to moderate against.

ModerationRuleType

Required. Indicates the type of rule to run on user-generated content.

Field Name	Field Type	Description
`type`	(enumeration of type string)	Required. Indicates the type of rule to run on user-generated content. Valid values are: • Content • Rate Available in API 39.0 and later.

RateLimitTimePeriod

Required. Indicates the time period that is applied to the rate limit.

Field Name	Field Type	Description
`timePeriod`	(enumeration of type string)	Required. Indicates the time period that is applied to the rate limit. Valid values are: • Short • Medium Available in API 39.0 and later.

Declarative Metadata Sample Definition

The following is an example of a ModerationRule component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ModerationRule xmlns="http://soap.sforce.com/2006/04/metadata">
   <description>Blocks Bad Word List in posts, comments, Link URLs, titles, and poll
choices.</description>
   <masterLabel>Blocking Rule</masterLabel>
   <action>Block</action>
   <active>true</active>
   <userMessage>You can't use %BLOCKED_KEYWORD% or other inappropriate words in this site.
 Review your content and try again.</userMessage>
   <!-- Applies the rule to FeedComment.RawCommentBody (an internal only field), if it
contains words from the keyword list specified -->
   <entitiesAndFields>
     <entityName>FeedComment</entityName>
     <fieldName>RawCommentBody</fieldName>
     <keywordList>site1.badword_list</keywordList>
   </entitiesAndFields>
   <entitiesAndFields>
     <entityName>FeedItem</entityName>
     <fieldName>LinkUrl</fieldName>
     <keywordList>site1.badword_list</keywordList>
   </entitiesAndFields>
  <!-- Applies the rule to FeedItem.RawBody (an internal only field), if it contains words
 from the keyword list specified -->
   <entitiesAndFields>
     <entityName>FeedItem</entityName>
     <fieldName>RawBody</fieldName>
     <keywordList>site1.badword_list</keywordList>
   </entitiesAndFields>
   <entitiesAndFields>
     <entityName>FeedItem</entityName>
     <fieldName>Title</fieldName>
     <keywordList>site1.badword_list</keywordList>
   </entitiesAndFields>
   <entitiesAndFields>
     <entityName>FeedPollChoice</entityName>
     <fieldName>ChoiceBody</fieldName>
     <keywordList>site1.badword_list</keywordList>
   </entitiesAndFields>
</ModerationRule>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
   <types>
     <name>ModerationRule</name>
     <members>site1.blocking_rule</members>
   </types>
   <version>36.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

MutingPermissionSet

Represents a set of disabled permissions and is used in conjunction with PermissionSetGroup.

This type extends the PermissionSet metadata type.

Declarative Metadata File Suffix and Directory Location

Muting permission sets are stored in the `mutingpermissionsets` directory. The file name matches the muting permission set API name and the extension is `.mutingpermissionset`. For example, a `mutingpermissionsets` with the name `Finance_Mgmt_MutingPermSet` is stored in `mutingpermissionsets/Finance_Mgmt_MutingPermSet.mutingpermissionset`.

Version

This object is available in API version 46.0 and later.

Special Access Rules

As of Summer '20 and later, only users who have one of these permissions can access this type:

- View Setup and Configuration
- Manage Session Permission Set Activations
- Assign Permission Sets
- Manage Profiles and Permission Sets

To view the following settings, assignments, and permissions for standard and custom objects in a specified muting permission set, the View Setup and Configuration permission is required.

- Client settings
- Field permissions
- Layout assignments
- Object permissions
- Permission dependencies
- Permission set tab settings
- Permission set group components
- Record types

Fields

MutingPermissionSet has the same fields as PermissionSet, plus a single field, `label`, used to name a MutingPermissionSet. Unlike PermissionSet, settings enabled by MutingPermissionSet are turned off for the permission set group that it's a component of.

Field	Field Type	Description
label	string	Required. The name of the muting permission set.

Declarative Metadata Sample Definition

The following example deploys a MutingPermissionSet used in a Permission Set Group intended for users submitting job applications for a custom application. The muting permission set has administrative permissions enabled to ensure that they're muted in the Permission Set Group.

```
<?xml version="1.0" encoding="UTF-8"?>
<MutingPermissionSet xmlns="http://soap.sforce.com/2006/04/metadata">
    <label>Job Apps User Muted</label>
    <description>Mutes any administrative tasks for the Job Apps user</description>
    <hasActivationRequired>false</hasActivationRequired>
    <license>Salesforce</license>
    <applicationVisibilities>
        <application>JobApps__Approval</application>
        <visible>true</visible>
    </applicationVisibilities>
    <classAccesses>
        <apexClass>ApprovalUtility</apexClass>
        <enabled>true</enabled>
    </classAccesses>
    <customPermissions>
        <enabled>true</enabled>
        <name>JobAppApprover</name>
    </customPermissions>
    <fieldPermissions>
        <editable>false</editable>
        <field>Job_Request__c.Salary__c</field>
        <readable>true</readable>
    </fieldPermissions>
    <objectPermissions>
        <allowCreate>true</allowCreate>
        <allowDelete>true</allowDelete>
        <allowEdit>true</allowEdit>
        <allowRead>true</allowRead>
        <customizeSetup>true</customizeSetup>
        <deleteSetup>true</deleteSetup>
        <modifyAllRecords>true</modifyAllRecords>
        <object>Approval_Confirmation__c</object>
        <viewAllRecords>true</viewAllRecords>
        <viewSetup>true</viewSetup>
    </objectPermissions>
    <pageAccesses>
        <apexPage>Job_Approval_Web_Form</apexPage>
        <enabled>true</enabled>
    </pageAccesses>
    <recordTypeVisibilities>
        <recordType>Approval_Confirmation__c.DevManager</recordType>
        <visible>true</visible>
    </recordTypeVisibilities>
```

```
    <tabSettings>
        <tab>Approval_Confirmation__c</tab>
        <visibility>Visible</visibility>
    </tabSettings>
</MutingPermissionSet>
```

The following is an example package.xml manifest used to retrieve the MutingPermissionSet metadata for an organization.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>Job_Apps_User</members>
        <name>PermissionSetGroup</name>
    </types>
    <types>
        <members>Job_Apps_User_Muted</members>
        <name>MutingPermissionSet</name>
    </types>
    <version>49.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SEE ALSO:

PermissionSet

MyDomainDiscoverableLogin

Represents the configuration settings when the My Domain login page type is Discovery. Login Discovery provides an identity-first login experience, where the login page contains the identifier field only. Based on the identifier entered, a handler determines how to authenticate the user. This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

MyDomainDiscoverableLogin components have the suffix `.myDomainDiscoverableLogin` in the `myDomainDiscoverableLogins` folder.

Version

MyDomainDiscoverableLogin components are available in API version 48.0 and later.

Fields

Field Name	Field Type	Description
apexHandler	string	Required. The Apex handler class that contains the Discovery authentication logic.
executeApexHandlerAs	string	The user who is executing the handler. Requires the Manage User permission.
usernameLabel	string	The login prompt when the My Domain login page type is Discovery. This label supports localization with custom labels.

Declarative Metadata Sample Definition

The following is an example of a MyDomainDiscoverableLogin component.

```
<?xml version="1.0" encoding="UTF-8"?>
<MyDomainDiscoverableLogin xmlns="http://soap.sforce.com/2006/04/metadata">
    <apexHandler>MyDomainDiscLoginHandler</apexHandler>
    <executeApexHandlerAs>executeUser@example.com</executeApexHandlerAs>
    <usernameLabel>Enter your email</usernameLabel>
</MyDomainDiscoverableLogin>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>MyDomainDiscoverableLogin</name>
    </types>
    <version>48.0</version>
</Package>
```

Usage

Use this type to access the My Domain Login Discovery Page. This type of login page prompts users to identity themselves with an email address, phone number, or custom identifier. My Domain Login Discovery performs an interview-based login process, where users are prompted to provide identity for authentication. For example, users receive a verification code that they enter to complete the login process.

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

NamedCredential

Represents a named credential, which specifies the URL of a callout endpoint and its required authentication parameters in one definition. A named credential can be specified as an endpoint to simplify the setup of authenticated callouts.

Note: All credentials stored within this entity are encrypted under a framework that is consistent with other encryption frameworks on the platform. Salesforce encrypts your credentials by auto-creating org-specific keys. Credentials encrypted using the previous encryption scheme have been migrated to the new framework.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

NamedCredential components have the suffix `.namedCredential` and are stored in the `namedCredentials` folder.

Version

NamedCredential components are available in API version 33.0 and later.

Special Access Rules

As of Spring '20 and later, only users with the View Setup and Configuration permission can access this type.

Fields

Field Name	Description
`allowMergeFieldsInBody`	**Field Type** boolean **Description** Specifies whether Apex code can use merge fields to populate the HTTP request body with org data when a callout is made. Corresponds to **Allow Merge Fields in HTTP Body** in the user interface. Defaults to `false`. This field is available in API version 41.0 and later.
`allowMergeFieldsInHeader`	**Field Type** boolean **Description** Specifies whether Apex code can use merge fields to populate the HTTP header with org data when a callout is made. Corresponds to **Allow Merge Fields in HTTP Header** in the user interface. Defaults to `false`. This field is available in API version 41.0 and later.

Field Name	Description
`authProvider`	**Field Type** string **Description** The authentication provider that the AuthProvider component represents. This field is valid only when NamedCredentialType is set to `Legacy`. **This field is deprecated in API version 56.0.**
`authTokenEndpointUrl`	**Field Type** string **Description** The URL where JWTs are exchanged for access tokens. This field is valid only when NamedCredentialType is set to `Legacy`. **First available in API version 46.0, this field is deprecated in API version 56.0 and later.**
`awsAccessKey`	**Field Type** string **Description** First part of the access key used to sign programmatic requests to AWS. Use when AWS Signature Version 4 is your authentication protocol. This field is valid only when NamedCredentialType is set to `Legacy`. **First available in API version 46.0, this field is deprecated in API version 56.0 and later.**
`awsAccessSecret`	**Field Type** string **Description** The second part of the access key used to sign programmatic requests to AWS. Use when AWS Signature Version 4 is your authentication protocol. This field is valid only when NamedCredentialType is set to `Legacy`. **First available in API version 46.0, this field is deprecated in API version 56.0 and later.**
`awsRegion`	**Field Type** string **Description** Specifies which AWS Region the named credential accesses. This field is valid only when NamedCredentialType is set to `Legacy`. **First available in API version 46.0, this field is deprecated in API version 56.0 and later.**

Field Name	Description
`awsService`	**Field Type** string **Description** Specifies which AWS resource the named credential accesses. This field is valid only when NamedCredentialType is set to `Legacy`. **First available in API version 46.0, this field is deprecated in API version 56.0 and later.**
`calloutStatus`	**Field Type** calloutStatus (enumeration of type string) **Description** Specifies whether the named credential is enabled for callouts. Valid values are: • `Disabled`: The named credential is disabled for callouts. • `Enabled`: The named credential is enabled for callouts. This field is available in API version 59.0 and later.
`certificate`	**Field Type** string **Description** If you specify a certificate, your Salesforce org supplies it when establishing each two-way SSL connection with the external system. The certificate is used for digital signatures, which verify that requests are coming from your Salesforce org. This field is valid only when NamedCredentialType is set to `Legacy`. **This field is deprecated in API version 56.0.**
`endpoint`	**Field Type** string **Description** The URL or root URL of the callout endpoint. Corresponds to **URL** in the user interface. This field is valid only when NamedCredentialType is set to `Legacy`. **This field is deprecated in API version 56.0.**
`generateAuthorizationHeader`	**Field Type** boolean **Description** Specifies whether Salesforce generates an authorization header and applies it to each callout that references the named credential. Corresponds to **Generate Authorization Header** in the user interface. Defaults to `true`. This field is available in API version 41.0 and later.

Field Name	Description
`jwtAudience`	**Field Type** string **Description** External service or other allowed recipients for the JWT. Written as JSON, with a quoted string for a single audience and an array of quoted strings for multiple audiences. Single audience example: “aud1” Multiple audiences example: [“aud1”, “aud2”, “aud3”]. This field is valid only when NamedCredentialType is set to `Legacy`. **This field is deprecated in API version 56.0.**
`jwtFormulaSubject`	**Field Type** string **Description** Formula string calculating the Subject of the JWT. API names and constant strings, in single quotes, can be included. Allows a dynamic Subject unique per user requesting the token. For example, *`'User='+$User.Id`*. Use this field when `principalType` is set to `PerUser`. Corresponds to `Per User Subject` in the user interface. This field is valid only when NamedCredentialType is set to `Legacy`. **First available in API version 46.0, this field is deprecated in API version 56.0 and later.**
`jwtIssuer`	**Field Type** string **Description** Specify who issued the JWT using a case-sensitive string. This field is valid only when NamedCredentialType is set to `Legacy`. **First available in API version 46.0, this field is deprecated in API version 56.0 and later.**
`jwtSigningCertificate`	**Field Type** string **Description** Certificate verifying the JWT’s authenticity to external sites. This field is valid only when NamedCredentialType is set to `Legacy`. **First available in API version 46.0, this field is deprecated in API version 56.0 and later.**
`jwtTextSubject`	**Field Type** string

Field Name	Description
	Description Static text, without quotes, that specifies the JWT Subject. Use this field when `principalType` is set to `NamedUser`. Corresponds to `Named Principal Subject` in the user interface. This field is valid only when NamedCredentialType is set to `Legacy`. **First available in API version 46.0, this field is deprecated in API version 56.0 and later.**
`jwtValidityPeriodSeconds`	**Field Type** int **Description** Specify the number of seconds that the token is valid. This field is valid only when NamedCredentialType is set to `Legacy`. **First available in API version 46.0, this field is deprecated in API version 56.0 and later.**
`label`	**Field Type** string **Description** Required. A user-friendly name for the named credential that appears in the Salesforce user interface, such as in list views.
`namedCredentialParameters`	**Field Type** NamedCredentialParameter[] **Description** Reference to the (one or more) NamedCredentialParameter used to configure a named credential. This field is available in API version 56.0 and later.
`namedCredentialType`	**Field Type** NamedCredentialType (enumeration of type string) **Description** Specifies the type or behavior of this named credential. Valid values are: • `Legacy`: The named credential is a legacy type, which means that it doesn't use the schema introduced in the Winter '23 release. Used for backward compatibility. • `PrivateEndpoint`: The named credential sends traffic through a private connection, bypassing the public internet. If the credential type is `PrivateEndpoint`, you must specify the value of `OutboundNetworkConnection`.

Field Name	Description
	• `SecuredEndpoint`: The named credential is extensible and uses external credentials to control authentication and permissions. • `Standard`: Reserved for internal use. This field is available in API version 56.0 and later.
`oauthRefreshToken`	**Field Type** string **Description** The OAuth refresh token. Used to obtain a new access token for an end user when a token expires. This field is valid only when NamedCredentialType is set to `Legacy`. **This field is deprecated in API version 56.0.**
`oauthScope`	**Field Type** string **Description** Specifies the scope of permissions to request for the access token. Corresponds to **Scope** in the user interface. This field is valid only when NamedCredentialType is set to `Legacy`. **This field is deprecated in API version 56.0.**
`oauthToken`	**Field Type** string **Description** The access token that's issued by your authorization server. This field is valid only when NamedCredentialType is set to `Legacy`. **This field is deprecated in API version 56.0.**
`outboundNetworkConnection`	**Field Type** string **Description** Specifies the outbound network connection that uses the named credential to send callouts to AWS. This field is valid only when NamedCredentialType is set to `Legacy`. **First available in API version 49.0, this field is deprecated in API version 56.0 and later.**
`password`	**Field Type** string

Field Name	Description
	Description The password to be used by your org to access the external system. Ensure that the credentials have adequate privileges to access the external system. Depending on how you set up access, you might need to provide the administrator password. This field is valid only when NamedCredentialType is set to `Legacy`. **This field is deprecated in API version 56.0.**
`principalType`	**Field Type** ExternalPrincipalType (enumeration of type string) **Description** Determines whether you're using one set or multiple sets of credentials to access the external system. Corresponds to **Identity Type** in the user interface. Values are: • `Anonymous` • `NamedUser` • `PerUser` This field is valid only when NamedCredentialType is set to `Legacy`. **This field is deprecated in API version 56.0.**
`protocol`	**Field Type** AuthenticationProtocol (enumeration of type string) **Description** The authentication protocol that's required to access the external system. Valid values are: • `AwsSv4` • `Jwt` • `JwtExchange` • `NoAuthentication` • `Oauth` • `Password` For connections to Amazon Web Services using Signature Version 4, use `AwsSv4`. For connections using a direct token system, select `Jwt`. If using an intermediary authorization provider to process JWTs and return access tokens, use `JwtExchange`. For Simple URL data sources, select `NoAuthentication`. For cloud-based Files Connect external systems, select `Oauth`. For on-premises systems, select `Password`. This field is valid only when NamedCredentialType is set to `Legacy`. **This field is deprecated in API version 56.0.**

Field Name	Description
username	**Field Type** string **Description** The username to be used by your org to access the external system. Ensure that the credentials have adequate privileges for performing callouts to the external system. Depending on how you set up access, you might need to provide the administrator username. This field is valid only when NamedCredentialType is set to `Legacy`. **This field is deprecated in API version 56.0.**

NamedCredentialParameter

Represents the parameters that configure a named credential. Named credential parameters are used to configure Named Credential callouts through a combination of the type, name, and value/lookup fields. Available in API version 56.0 and later.

These parameters are used internally to provide a flexible architecture and are exposed here for packaging reasons.

Field Name	Description
certificate	**Field Type** string **Description** If the value of the `parameterType` field is `ClientCertificate` then this field references the certificate.
description	**Field Type** string **Description** A human-readable description of this named credential parameter.
externalCredential	**Field Type** string **Description** If the value of the `parameterType` field is `Authentication`, then this field references an external credential that in turn references a set of authenticated user credentials.
globalNamedPrincipalCredential	**Field Type** boolean **Description** Reserved for internal use.

Field Name	Description
managedFeatureEnabledCallout	**Field Type** boolean **Description** Reserved for internal use.
outboundNetworkConnection	**Field Type** string **Description** The lookup field for the OutboundNetworkConnection parameter type. Used when namedCredentialType is PrivateEndpoint.
parameterName	**Field Type** string **Description** Required. The name of the named credential parameter.
parameterType	**Field Type** NamedCredentialParamType (enumeration of type string) **Description** Required. The type of the named credential parameter. Valid values are: • AllowedManagedPackageNamespaces: Allows managed packages identified by specified namespaces to use the named credential and make callouts through it. • Authentication: Specifies that this parameter configures authentication using the credentials specified in the external credential, referenced by the externalCredential field. • ClientCertificate: Specifies that this parameter configures a client certificate, referenced by the certificate field. • ConnectionStatus: Reserved for internal use. • CreatedByNamespace: Reserved for internal use. • CustomParameter: Reserved for internal use. • HttpHeader: Allows the user to specify custom headers to be added to the callout at run time. When using HttpHeader, the parameterName field must be the header name as a string, and parameterValue must be a formula of a header value that is evaluated at run time. • ManagedByComponent: Reserved for internal use. • ManagedByFeature: Reserved for internal use. • OutboundNetworkConnection: Specifies a lookup to an outbound network connection. When using this parameter type, the

Field Name	Description
	`outboundNetworkConnection` field is a string representing the lookup. Used when `namedCredentialType` is `PrivateEndpoint`. • `StandardNamedCredentialType`: Reserved for internal use. • `Url`: Specifies that this parameter configures the URL of the endpoint. Store the actual URL in the `parameterValue` field.
`parameterValue`	**Field Type** string **Description** If the `parameterType` field describes a literal value, such as `Url`, then the literal value is stored in this field, such as `https://iam.amazonaws.com/`.
`readOnlyNamedCredential`	**Field Type** boolean **Description** Reserved for internal use.
`sequenceNumber`	**Field Type** int **Description** Used to order `HttpHeader` parameters.
`systemUserNamedCredential`	**Field Type** boolean **Description** Reserved for internal use.

Declarative Metadata Sample Definition

The following is an example of a NamedCredential component.

```
<?xml version="1.0" encoding="UTF-8"?>
<NamedCredential xmlns="http://soap.sforce.com/2006/04/metadata">
    <label>SampleNamedCredential</label>
    <namedCredentialType>SecuredEndpoint</namedCredentialType>
    <namedCredentialParameters>
        <description>IAM Endpoint</description>
        <parameterName>DefaultEndpoint</parameterName>
        <parameterType>Url</parameterType>
        <parameterValue>https://iam.amazonaws.com/</parameterValue>
    </namedCredentialParameters>
    <namedCredentialParameters>
        <description>AWS Auth</description>
        <parameterName>DefaultAuth</parameterName>
        <parameterType>Authentication</parameterType>
```

```
        <externalCredential>SampleExternalCredential</externalCredential>
    </namedCredentialParameters>
    <namedCredentialParameters>
        <description>Cert</description>
        <parameterName>DefaultCert</parameterName>
        <parameterType>ClientCertificate</parameterType>
        <certificate>MyCertificate</certificate>
    </namedCredentialParameters>
    <allowMergeFieldsInBody>true</allowMergeFieldsInBody>
    <allowMergeFieldsInHeader>true</allowMergeFieldsInHeader>
    <generateAuthorizationHeader>true</generateAuthorizationHeader>
</NamedCredential>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>NamedCredential</name>
    </types>
    <version>56.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SEE ALSO:

- ExternalCredential
- *Salesforce Help*: Named Credentials
- *Named Credentials Developer Guide*: Get Started with Named Credentials
- *Named Credentials Developer Guide*: Named Credential API Links
- *Apex Developer Guide*: Invoking Callouts Using Apex
- *Apex Developer Guide*: Named Credentials as Callout Endpoints

NavigationMenu

Represents the navigation menu in an Experience Builder site. A navigation menu consists of items that users can click to go to other parts of the site. This type replaces the NavigationLinkSet subtype on Network. NavigationMenu is available in API version 47.0 and later. This type extends the Metadata metadata type and inherits its `fullName` field.

The Help Center and LWR templates (Build Your Own and Microsites) don't include generic record pages. So if you create an object or global action type menu item that links to a Salesforce object, make sure that you also create the corresponding object pages. If you don't create the associated object pages, end users won't see anything if they click on the menu item.

File Suffix and Directory Location

NavigationMenu components have the suffix `.navigationMenu` and are stored in the `navigationMenus` folder.

Version

NavigationMenu components are available in API version 47.0 and later.

Special Access Rules

The MultipleNavigationMenu permission is required.

Fields

Field	Field Type	Description
`container`	string	The name of the navigation menu container.
`containerType`	string	The container type. The options are Network or CommunityTemplateDefinition.
`label`	string	The navigation menu label as it appears in the Experience Builder UI.
`navigationMenuItem`	NavigationMenuItem[]	A list of menu items in a NavigationMenu. Use this object to create, delete, or update menu items in your site's navigation menu.

NavigationMenuItem

Represents a single menu item in the NavigationLinkSet subtype on Network (API version 37.0 to 46.0) or in the NavigationMenu type (API version 47.0 and later). Use this object to create, delete, or update menu items in your site's navigation menu.

Field	Field Type	Description
`defaultListViewId`	string	If the value of the `type` field is `SalesforceObject`, the value is the ID of the default list view for the object.
`label`	string	Required. The text that appears in the navigation menu for this item.
`menuItemBranding`	NavigationMenuItemBranding	Branding for the navigation menu item. Available in API version 47.0 and later.
`position`	int	Required. The location of the menu item in the navigation menu.
`publiclyAvailable`	boolean	When set to `true`, gives access to guest users.

Field	Field Type	Description
`subMenu`	NavigationSubMenu	A list of child menu items. This field is available in API 39.0 and later.
`target`	string	Required if `type` is `ExternalLink`, `InternalLink`, or `SalesforceObject`. If `type` is `ExternalLink` or `InternalLink`, the target is the URL that the link points to. For `ExternalLink`, your entry looks like this: *`https://salesforce.com`*. For `InternalLink`, use a relative URL, such as */`contactsupport`*. If `type` is `MenuLabel` or `NavigationalTopic`, `target` isn't used.
`targetPreference`	string	Backed by a picklist that includes preferences for the target field. Valid values are: • `None` • `OpenInExternalTab`—Used for external links to determine whether to open in an external tab.
`type`	string	Required. The type of navigation menu item. Valid values are: • `SalesforceObject`—Available objects include accounts, cases, contacts, and custom objects. • `ExternalLink`—Links to a URL outside of your site. For example, *`https://salesforce.com`*. • `InternalLink`—Links to a relative URL inside your site. For example, */`contactsupport`*. • `MenuLabel`—A parent heading for your navigation menu. See NavigationSubMenu for how to nest items underneath the menu label. This value is available in API 39.0 and later. • `NavigationalTopic`—A dropdown list with links to the navigational topics in your site. You can't nest other items of type `MenuLabel` or

Field	Field Type	Description
		`NavigationalTopic` under `MenuLabel`.

NavigationMenuItemBranding

Branding for a menu item.

Field	Field Type	Description
`tileImage`	string	Name of the ContentAsset to use for the navigation menu item.

NavigationSubMenu

A list of child menu items. Only NavigationMenuItem items of type `MenuLabel` can have items in a NavigationSubMenu. Available in API 39.0 and later.

Field	Field Type	Description
`navigationMenuItem`	NavigationMenuItem[]	A list of menu items in a NavigationSubMenu. Use `navigationMenuItem` to create, delete, or update child items under a parent heading.

Declarative Metadata Sample Definition

The following is an example of a NavigationMenu component.

```
<?xml version="1.0" encoding="UTF-8"?>
<NavigationMenu xmlns="http://soap.sforce.com/2006/04/metadata">
    <container>Service</container>
    <containerType>Network</containerType>
    <label>Test Navigation</label>
    <navigationMenuItem>
        <label>Accounts</label>
        <position>1</position>
        <publiclyAvailable>false</publiclyAvailable>
        <target>Account</target>
        <type>SalesforceObject</type>
    </navigationMenuItem>
    <navigationMenuItem>
        <label>External Link</label>
        <menuItemBranding>
            <tileImage>google_image</tileImage>
        </menuItemBranding>
```

```
        <position>2</position>
        <publiclyAvailable>false</publiclyAvailable>
        <target>http://google.com</target>
        <targetPreference>OpenExternalLinkInSameTab</targetPreference>
        <type>ExternalLink</type>
    </navigationMenuItem>
    <navigationMenuItem>
        <label>All Objects</label>
        <position>3</position>
        <publiclyAvailable>false</publiclyAvailable>
        <subMenu>
            <navigationMenuItem>
                <label>Leads</label>
                <position>0</position>
                <publiclyAvailable>false</publiclyAvailable>
                <target>Account</target>
                <type>SalesforceObject</type>
            </navigationMenuItem>
        </subMenu>
        <type>MenuLabel</type>
    </navigationMenuItem>
</NavigationMenu>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
  <types>
    <members>*</members>
    <name>NavigationMenu</name>
  </types>
  <version>47.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

Network

Represents an Experience Cloud site. Salesforce Experience Cloud lets you create branded spaces for your employees, customers, and partners. You can customize and create experiences, whether they're communities, sites, or portals, to meet your business needs, then transition seamlessly between them. If you want to create zones that contain Chatter Answers and Ideas, use the Community (Zone) component.

This type extends the Metadata metadata type and inherits its `fullName` field.

Declarative Metadata File Suffix and Directory Location

Network components are stored in the `networks` directory of the corresponding package directory. The file name matches the site name, and the extension is `.network`.

Version

This object is available in API version 28.0 and later.

Fields

Field	Field Type	Description
`allowedExtensions`	string	Specifies the types of files allowed in your site. This list of file types lets you control what your members upload and also prevents spammers from polluting your site with inappropriate files. Available in API version 36.0 and later.
`allowInternalUserLogin`	boolean	Determines whether internal users can log in with their internal credentials on the site login page. Available in API version 40.0 and later.
`allowMembersToFlag`	boolean	Determines whether users in the site can flag posts or comments as inappropriate. Flagged items are sent to a moderator for review. Available in API version 29.0 and later.
`branding`	Branding	The color scheme, header, and footer used in the site. Deprecated in API version 41.0 and later. Replaced by the NetworkBranding type.
`caseCommentEmailTemplate`	string	Email template used when notifying members when a case comment has been modified or added to a case. Lightning email templates aren't packageable. We recommend using a Classic email template.
`changePasswordTemplate`	string	Email template used when notifying a user that their password has been reset. Lightning email templates aren't packageable. We recommend using a Classic email template.
`chgEmailVerNewTemplate`	string	Email template used to verify a user's email address change. This email is sent to the new email address. **Note:** You can't update this template via metadata API.
`chgEmailVerOldTemplate`	string	Email template used to verify a user's email address change. This email is sent to the old email address. **Note:** You can't update this template via metadata API.
`communityRoles`	CommunityRoles	Identifies users with Customer, Partner, or Employee roles in a site. Available in API version 41.0 and later.

Field	Field Type	Description
description	string	Description of the site.
deviceActEmailTemplate	string	The ID of the device activation email template. The template is used to customize the device activation email for community users. Available in API version 53.0 and later.
disableReputationRecordConversations	boolean	When reputation levels are enabled for the site, determines whether to exclude contributions to records when counting points toward reputation levels. Available in API version 41.0 and later.
emailFooterLogo	string	The document name of the logo that appears in the footer of emails. Available in API version 41.0 and later.
emailFooterText	string	The text that appears in the footer of emails. Available in API version 41.0 and later.
emailSenderAddress	string	Required. Email address from which emails are sent.
emailSenderName	string	Required. Name from which emails are sent.
enableApexCDNCaching	boolean	Determines whether public data from @wire calls to Apex methods is cached for guest users. This setting applies only to sites using Salesforce's CDN for Digital Experiences.
enableCustomVFErrorPageOverrides	boolean	Determines whether to use custom Visualforce error pages instead of the default Visualforce error pages. Available in API version 41.0 and later.
enableDirectMessages	boolean	Determines whether site users can send direct messages to start a private conversation with one or more members. Available in API version 41.0 and later.
enableExperienceBundleBasedSnaOverrideEnabled	boolean	Determines whether the Builder-based SNA page is used (`true`) or not (`false`) and overrides the existing SNA page when an experience is published. Available in API version 52.0 and later.
enableGuestChatter	boolean	Specifies whether guest users can access public Chatter groups in the site without logging in.
enableGuestFileAccess	boolean	Determines whether guest users view asset files shared with the site on publicly accessible pages and login pages. If public access is enabled in Experience Builder at the page or site level, this property is automatically enabled. Available in API version 41.0 and later.
enableGuestMemberVisibility	boolean	Determines if unauthenticated guest users can see the authenticated members (`true`) or not (`false`). Available in API version 47.0 and later.

Field	Field Type	Description
`enableImageOptimizationCDN`	boolean	Determines whether to optimize cached images for guest users on all devices when a site uses Salesforce's CDN for Digital Experiences. Available in API version 56.0 and later.
`enableInvitation`	boolean	Determines whether users can invite others to the site.
`enableKnowledgeable`	boolean	Determines if members can see who's knowledgeable on topics and endorse people for their knowledge on a topic. Available in API version 30.0 and later.
`enableMemberVisibility`	boolean	Controls user visibility on a per-site basis. If `true`, the See other members of this site preference is enabled for the selected site. Available in API version 45.0 and later.
`enableNicknameDisplay`	boolean	Determines if user nicknames display instead of their first and last names in most places in the site. Set to `false` by default. Available in API version 32.0 and later.
`enablePrivateMessages`	boolean	Determines if members can send and receive private messages. Available in API version 30.0 and later.
`enableReputation`	boolean	Determines if reputation is calculated and displayed for members. Available in API version 31.0 and later. If enabled, `reputationLevels` and `reputationPointsRules` are used. If no `reputationLevels` or `reputationPointsRules` are defined in the data file, the default values are used.
`enableShowAllNetworkSettings`	boolean	Shows settings that are hidden by default based on how the site is set up. Available in API version 41.0 and later.
`enableSiteAsContainer`	boolean	Determines whether the site is an Experience Builder site (`true`) or a Salesforce Tabs + Visualforce site (`false`).
`enableTalkingAboutStats`	boolean	Determines whether users see how many people are discussing a topic. The number of people discussing the topic appears as the user types the topic and the system gives topic suggestions. Available in API version 41.0 and later.
`enableTopicAssignmentRules`	boolean	Enables the org to use rules to automatically assign topics to articles in a site. After it's enabled, admins set up rules to map topics to Salesforce Knowledge data categories. This field is available in API version 40.0 and later.
`enableTopicSuggestions`	boolean	Enables topic suggestions when users write posts. Available in API version 41.0 and later.
`enableUpDownVote`	boolean	Replaces the option to like a question or answer with the option to upvote or downvote. Available in API version 41.0 and later.

Field	Field Type	Description
`expFriendlyUrlsAsDefault`	boolean	Determines whether URL slugs are enabled by default on • Product and Category pages of LWR Commerce stores (available in API version 58.0 and later) • Custom object pages on enhanced LWR sites (available in API version 60.0 and later) • Account and contact pages on enhanced LWR sites (available in API version 61.0 and later)
`feedChannel`	string	Displays the feed of all channel program record or group interactions, including posts, questions, and attachments. This field is available in API version 28.0 and later.
`forgotPasswordTemplate`	string	Required. The email template used when a user forgets their password. Lightning email templates aren't packageable. We recommend using a Classic email template.
`gatherCustomerSentimentData`	boolean	Gathers data when a customer looks at articles and cases in sites, for use in the Community 360 feature. This field is available in API version 40.0 and later.
`lockoutTemplate`	string	The email template used to communicate with users when they get locked out of their org because of too many failed login attempts. Available in API version 43.0 and later. Lightning email templates aren't packageable. We recommend using a Classic email template.
`logoutUrl`	string	Specifies the URL that members are redirected to when they log out from your site. This field is available in API version 28.0 and later.
`maxFileSizeKb`	int	Specifies the maximum file size (in KBs) that members can upload in your site. Available in API version 36.0 and later. Enter a number between 3072 KB and your org's maximum file size. To use the default limit of 2 GB, leave this field empty.
`navigationLinkSet`	NavigationLinkSet	Represents the navigation menu in a site. A navigation menu consists of items that users can click to go to other parts of the site. This field is available in API versions 37.0 to 46.0. In API version 47.0 and later, use the NavigationMenu type instead.
`networkAuthApiSettings`	NetworkAuthApiSettings	The settings that control enablement, access, and security for the Headless Registration Flow, Headless Forgot Password Flow, Headless Passwordless Login Flow, and their associated APIs. Available in API version 60.0 and later.

Field	Field Type	Description
networkMemberGroups	NetworkMemberGroups	The profiles and permission sets that have access to the site. Users with these profiles or permission sets are members of the site. Note: If a Chatter customer (from a customer group) is assigned a permission set that is also associated with a site, the Chatter customer isn't added to the site.
networkPageOverrides	NetworkPageOverride	The settings in the Administration area (in Experience Management or Experience Workspaces) that control which page type Change Password, Forgot Password, Home, and Login each point to. Available in API version 40.0 and later.
newSenderAddress	string	Email address that has been entered as the new value for EmailSenderAddress but hasn't been verified yet. After a user has requested to change the sender email address and has successfully responded to the verification email, the NewSenderAddress value overwrites the value in EmailSenderAddress. This value becomes the email address from which emails are sent.
picassoSite	string	Name of the site of ChatterNetworkPicasso type that's linked to the Experience Cloud site.
recommendationAudience	RecommendationAudience	Creates an audience of new members, or can be used to manage customized lists of audience members to organize and target recommendations. Available in API version 41.0 and later.
recommendationDefinition	RecommendationDefinition	Represents a custom recommendation to drive engagement. Targets a specific audience and uses channels to specify a location for the recommendation. Available in API version 41.0 and later.
reputationLevels	ReputationLevelDefinitions	The reputation levels assigned to members when they accrue points by performing certain actions.
reputationPointsRules	ReputationPointsRules	The points members accrue when they perform certain defined actions.
selfRegMicroBatchSubErrorEmailTemplate	reference	The email template used to communicate with users when their self-registration request, using micro-batching failed. Available in API version 54.0 and later.
selfRegProfile	string	The profile assigned to users who self-register. This value is used only if selfRegistration is enabled for the site. Available in API version 29.0 and later.
selfRegistration	boolean	Determines whether self-registration is available for the site.

Field	Field Type	Description
`sendWelcomeEmail`	boolean	Determines whether a welcome email is sent when a new user is added to the site.
`site`	string	Required. The CustomSite associated with the Experience Cloud site.
`siteArchiveStatus`	SitesArchiveStatus	Specifies whether the site has been archived. Available values are: • `NotArchived`—The site hasn't been archived. • `TemporarilyArchived`—The site is archived, but can be unarchived in the future.
`status`	NetworkStatus[]	Required. Status of the site. Available values are: • `Live`—The site is online and members can access it. • `DownForMaintenance`—The site was previously published but was taken offline. Members with the Create and Set Up Experiences permission can still access the setup for offline sites regardless of profile or membership. Members aren't able to access offline sites, but they still appear in the user interface dropdown as `SiteName (Offline)`. • `UnderConstruction`—The site hasn't yet been published. Users with the Create and Set Up Experiences permission can access sites in this status if their profile is associated with the site. After a site is published, it can never be in this status again.
`tabs`	NetworkTabSet	Required. The tabs that are available in the site. The user that created the site selected these tabs.
`urlPathPrefix`	string	The first part of the path on the site's URL that distinguishes this site from other sites. For example, if your site URL is *`MyDomainName`*`.my.site.com/partners`, then `partners` is the `urlPathPrefix`.
`verificationTemplate`	string	The email template used to communicate with users when they must verify their identity, for example, when they log in without a password or from a new device. Available in API version 44.0 and later. Lightning email templates aren't packageable. We recommend using a Classic email template.

Field	Field Type	Description
`welcomeTemplate`	string	The email template used when sending welcome emails to new members. Lightning email templates aren't packageable. We recommend using a Classic email template.

Branding

Represents the branding and color scheme applied to the Experience Cloud site. Available in API version 40.0 and earlier. Replaced by NetworkBranding in API version 41.0 and later.

Field	Field Type	Description
`loginFooterText`	string	The text that appears in the footer of the login page.
`loginLogo`	string	The logo that appears on the login page for external users.
`pageFooter`	string	An image that appears on the footer of the pages. Must be an .html file.
`pageHeader`	string	An image that appears on the header of the pages. Can be an .html, .gif, .jpg, or .png file.
`primaryColor`	string	The color used for the active tab.
`primaryComplementColor`	string	Font color used with `primaryColor`.
`quaternaryColor`	string	The background color for pages.
`quaternaryComplementColor`	string	Font color used with `quaternaryColor`.
`secondaryColor`	string	The color used for the top borders of lists and tables.
`tertiaryColor`	string	The background color for section headers on edit and detail pages.
`tertiaryComplementColor`	string	Font color used with `tertiaryColor`.
`zeronaryColor`	string	The background color for the header.
`zeronaryComplementColor`	string	Font color used with `zeronaryColor`.

CommunityRoles

The labels used to identify users with Customer, Partner, or Employee roles in an Experience Cloud site. Available in API version 41.0 and later.

Field	Field Type	Description
customerUserRole	string	The label for the Customer user role.
employeeUserRole	string	The label for the Employee user role.
partnerUserRole	string	The label for the Partner user role.

NavigationLinkSet

Represents the navigation menu in an Experience Cloud site. A navigation menu consists of items that users can click to go to other parts of the site. Available in API versions 37.0 to 46.0. In API version 47.0, use NavigationMenu instead.

Field	Field Type	Description
navigationMenuItem	NavigationMenuItem[]	A list of menu items in a NavigationLinkSet. Use this object to create, delete, or update menu items in your site's navigation menu.

NavigationMenuItem

Represents a single menu item in the NavigationLinkSet subtype (API version 37.0 to 46.0) or in the NavigationMenu type (API version 47.0 and later). Use this subtype to create, delete, or update menu items in your site's navigation menu.

Field	Field Type	Description
defaultListViewId	string	If the value of the `type` field is `SalesforceObject`, the value is the ID of the default list view for the object.
label	string	Required. The text that appears in the navigation menu for this item.
menuItemBranding	NavigationMenuItemBranding	Branding for the navigation menu item. Available in API version 47.0 and later.
position	int	Required. The location of the menu item in the navigation menu.
publiclyAvailable	boolean	When set to `true`, gives access to guest users.
subMenu	NavigationSubMenu	A list of child menu items. This field is available in API 39.0 and later.
target	string	Required if `type` is `ExternalLink`, `InternalLink`, or `SalesforceObject`. If `type` is `ExternalLink` or `InternalLink`, the target is the URL that the link points to. For `ExternalLink`, your entry looks like this: *`https://salesforce.com`*. For

Field	Field Type	Description
		`InternalLink`, use a relative URL, such as `/contactsupport`. If `type` is `MenuLabel` or `NavigationalTopic`, `target` isn't used.
`targetPreference`	string	Backed by a picklist that includes preferences for the target field. Valid values are: • `None` • `OpenInExternalTab`—Used for external links to determine whether to open in an external tab.
`type`	string	Required. The type of navigation menu item. Valid values are: • `SalesforceObject`—Available objects include accounts, cases, contacts, and custom objects. • `ExternalLink`—Links to a URL outside of your site. For example, `https://salesforce.com`. • `InternalLink`—Links to a relative URL inside your site. For example, `/contactsupport`. • `MenuLabel`—A parent heading for your navigation menu. See NavigationSubMenu for how to nest items underneath the menu label. This value is available in API 39.0 and later. • `NavigationalTopic`—A dropdown list with links to the navigational topics in your site. You can't nest other items of type `MenuLabel` or `NavigationalTopic` under `MenuLabel`.

NavigationSubMenu

A list of child menu items. Only NavigationMenuItem items of type `MenuLabel` can have items in a NavigationSubMenu. Available in API 39.0 and later.

Field	Field Type	Description
navigationMenuItem	NavigationMenuItem[]	A list of menu items in a NavigationSubMenu. Use navigationMenuItem to create, delete, or update child items under a parent heading.

NetworkAuthApiSettings

Represents the settings that control enablement, access, and security for the Headless Registration Flow, Headless Forgot Password Flow, Headless Passwordless Login Flow, and their associated APIs. Available in API version 60.0 and later.

Field	Details
doesForgotPasswordRequireAuth	Determines whether authentication is required to access Headless Forgot Password API when a password reset is requested. If true, an access token issued to an internal integration user in your initial POST request to the /services/auth/headless/forgot_password endpoint is required. The access token must include the forgot_password scope.
doesPasswordLoginRequireAuth	Determines whether reCAPTCHA is required for headless username-password login that uses the OAuth 2.0 for First-Party Applications draft protocol.
doesPwdlessLoginRequireAuth	Determines whether authentication is required to access Headless Passwordless Login API when user information is submitted to Salesforce. If true, an access token issued to an internal integration user is required in your initial POST request to the /services/auth/headless/init/passwordless/login endpoint. The access token must include the pwdless_login_api scope.
doesRegistrationRequireAuth	Determines whether authentication is required to access Headless Registration API when user registration information is submitted to Salesforce. If true, an access token issued to an internal integration user in your initial POST request to the /services/auth/headless/init/registration endpoint is required. The access token must include the user_registration_api scope.
emailTmplsAllowlist	The email template allowlist for the Headless Registration Flow, Headless Passwordless Login Flow, and Headless Forgot Password Flow. The allowlist defines which email templates can be used for verification emails sent to end users during these flows.
headlessDiscoveryExecutionUser	An integration user account to run a headless user discovery Apex handler.
headlessDiscoveryHandler	An Apex class that implements the Auth.HeadlessUserDiscoveryHandler interface.
isFirstPartyAppsAllowed	Determines whether the Experience Cloud site can use headless identity flows that use the OAuth 2.0 for First-Party Applications draft protocol.
isForgotPwdAllowed	Determines whether the Headless Forgot Password Flow is enabled.
isForgotPwdEmailTemplateAllowlistingEnabled	Determines whether email template allowlisting is enabled for the Headless Forgot Password Flow. If true, the emailtemplate parameter in the initial request to Headless Forgot Password API can include only allowlisted email templates.

Field	Details
isHeadlessUserRegistrationAllowed	Determines whether the Headless Registration Flow is enabled.
IsPwdlessLoginAllowed	Determines whether the Headless Passwordless Login Flow is enabled (`true`) or not (`false`).
isRecaptchaRequiredForgotPwd	Determines whether a reCAPTCHA token is required to access Headless Forgot Password API when a password reset is requested. If `true`, a reCAPTCHA token is required in your initial POST request to the `/services/auth/headless/forgot_password` endpoint.
isRecaptchaRequiredPwdlessLogin	Determines whether a reCAPTCHA token is required to access Headless Passwordless Login API when user information is submitted to Salesforce. If `true`, a reCAPTCHA token is required in your initial POST request to the `/services/auth/headless/init/passwordless/login` endpoint.
isRecaptchaRequiredRgstr	Determines whether a reCAPTCHA token is required to access Headless Registration API when user registration information is submitted to Salesforce. If `true`, a reCAPTCHA token is required in your initial POST request to the `/services/auth/headless/init/registration` endpoint.
isUniversalClientRgstrAllowed	Determines whether self-registration and passwordless login via Universal Registration API are enabled.
isUserDisambiguationAllowedForgotPwd	Determines whether the Headless Forgot Password Flow uses the headless user discovery Apex handler that's specified in the `HeadlessDiscoveryHandlerId` field. The handler enables users to reset their password with an identifier other than their username, such as an email address, phone number, or order number.
isUserDisambiguationAllowedUsernamePwd	Determines whether headless login flows use the headless user discovery Apex handler that's specified in the `HeadlessDiscoveryHandlerId` field. The handler enables users to log in with an identifier other than their username, such as an email address, phone number, or order number. This field applies to the Authorization Code and Credentials Flow and the OAuth 2.0 for First-Party Applications login flow.
maxPasswordResetAttempts	The maximum number of password reset attempts you allow for the Headless Forgot Password Flow before the user must request a new one-time password (OTP).
recaptchaScoreThreshold	The lowest reCAPTCHA score that is accepted before rejecting a request to access Headless Identity APIs. This value must be between 0.5 and 1. Scores closer to 0.5 are more likely to be bots, while scores closer to 1 are more likely to be valid users. You must set a score threshold if `doesForgotPasswordRequireAuth` or `doesRegistrationRequireAuth` fields are set to `true`. reCAPTCHA settings apply to both the Headless Registration Flow and the Headless Forgot Password Flow. Google issues a reCAPTCHA score only for reCAPTCHA v3 implementations. If you implement reCAPTCHA v2, this field doesn't apply.
recaptchaSecretKey	The reCAPTCHA secret key from your API key pair. You get the API key pair from Google when you set up reCAPTCHA. The secret key helps your app securely communicate with Google. You must enter a secret key if `doesForgotPasswordRequireAuth` or `doesRegistrationRequireAuth` are set to `true`. reCAPTCHA settings apply to all headless identity flows for which reCAPTCHA is enabled.

Field	Details
registrationExecutionUser	The user who runs your headless registration Apex handler.
registrationHandler	The headless registration Apex handler.
registrationUserDefaultProfile	The default profile that gets assigned to new users when they register.

NetworkEmailTmplAllowlist

Represents the allowlist for one-time password (OTP) email templates sent to end users during the Headless Registration Flow, Headless Passwordless Login Flow, and Headless Forgot Password Flow. Available in API version 60.0 and later.

Field	Field Type	Description
emailTemplate	string	Required. The email templates that can be sent to users during the headless authorization flows for registration, passwordless login, and forgot password. You can list multiple templates. When your app sends its initial request to Headless Registration API or Headless Passwordless Login API, the emailtemplate parameter can include only an email template ID from the allowlist. For Headless Forgot Password API, it works the same way, but only if the isForgotPwdEmailTemplateAllowlistingEnabled field on the NetworkAuthApiSettings metadata type is true.

NetworkMemberGroup

Represents the profiles and permission sets that are assigned to the Experience Cloud site. Users with one of the profiles or permission sets are members of the site, unless the user is a Chatter customer (from a customer group).

Field	Field Type	Description
permissionSet	string	A permission set that is assigned to the site. Note: If a Chatter customer (from a customer group) is assigned a permission set that is also associated with a site, the Chatter customer isn't added to the site.
profile	string	A profile that is part of the site.

NetworkPageOverride

Represents settings in the Administration area (in Experience Management or Experience Workspaces) that control which page type the Change Password, Forgot Password, Home, and Login pages each point to.

Note: Assigned Visualforce page overrides are specified and deployed via the corresponding CustomSite metadata field.

Field	Field Type	Description
`changePasswordPageOverrideSetting`	NetworkPageOverrideSetting (enumeration of type string)	Required. Specifies the page type that the Change Password page setting applies to. The valid values are: • `Configurable`—a configurable self-registration page • `Designer`—an Experience Builder page • `Standard`—the default page • `VisualForce`—a Visualforce page
`forgotPasswordPageOverrideSetting`	NetworkPageOverrideSetting (enumeration of type string)	Required. Specifies the page type that the Forgot Password page setting applies to. The valid values are: • `Configurable`—a configurable self-registration page • `Designer`—an Experience Builder page • `Standard`—the default page • `VisualForce`—a Visualforce page
`homePageOverrideSetting`	NetworkPageOverrideSetting (enumeration of type string)	Required. Specifies the page type that the Experience Home page setting applies to. The valid values are: • `Configurable`—a configurable self-registration page • `Designer`—an Experience Builder page • `Standard`—the default page • `VisualForce`—a Visualforce page
`loginPageOverrideSetting`	NetworkPageOverrideSetting (enumeration of type string)	Required. Specifies the page type that the Login page setting applies to. The valid values are: • `Configurable`—a configurable self-registration page • `Designer`—an Experience Builder page

Field	Field Type	Description
		• `Standard`—the default page • `VisualForce`—a Visualforce page Note: To configure an Experience Builder page for your Home and Login pages, make sure you publish your site. Unpublished pages show up as Default Page from the dropdown menu in Admin settings.
`selfRegProfilePageOverrideSetting`	NetworkPageOverrideSetting (enumeration of type string)	Required. Specifies the page type that the Self Registration page setting applies to. The valid values are: • `Configurable`—a configurable self-registration page • `Designer`—an Experience Builder page • `Standard`—the default page • `VisualForce`—a Visualforce page

RecommendationAudience

Creates an audience of new Experience Cloud site members, or can be used to manage customized lists of audience members to organize and target recommendations. Available in API version 41.0 and later.

Field	Field Type	Description
`recommendationAudienceDetails`	RecommendationAudienceDetail	The specific details of an audience for recommendations.

RecommendationAudienceDetail

The specific details of an audience for recommendations. Available in API version 41.0 and later.

Field	Field Type	Description
`audienceCriteriaType`	AudienceCriteriaType (enumeration of type string)	The criteria for the recommendation audience type. Values are: • `CustomList` • `MaxDaysInCommunity`
`audienceCriteriaValue`	string	For new member criteria, the maximum number of days since a user became a member. Null in case of custom list criteria.

Field	Field Type	Description
setupName	string	Name of the recommendation audience.

RecommendationDefinition

Represents a list of custom recommendations to drive engagement for an Experience Cloud site. Available in API version 41.0 and later.

Field	Field Type	Description
recommendationDefinitionDetails	RecommendationDefinitionDetail[]	A list of custom recommendations and their details.

RecommendationDefinitionDetail

The specific details of a custom recommendation. Available in API version 41.0 and later.

Field	Field Type	Description
actionUrl	string	The URL for the button that lets users act on the recommendation.
description	string	An explanation of the recommendation that suggests what users can do.
linkText	string	The text label for the button.
scheduledRecommendations	ScheduledRecommendation	A list of scheduled recommendations.
setupName	string	The name of the recommendation, which appears in Setup.
title	string	The title of the recommendation.

ReputationBranding

Branding for the reputation level.

Field	Field Type	Description
smallImage	string	Custom image associated with a reputation level. Use files with these extensions: .jpeg, .png, or .gif. Images are stored as documents. If not specified, the default reputation level image is used. Available in API version 32.0 and later.

ReputationLevelDefinitions

Represents reputation levels members can achieve by performing certain defined actions in an Experience Cloud site.

Field	Field Type	Description
`level`	ReputationLevel[]	Represents reputation levels.

ReputationLevel

Represents the name and lower value of the reputation level. The application calculates the upper value.

Field	Field Type	Description
`branding`	ReputationBranding[]	Represents any branding associated with the reputation level, specifically, the custom image for the reputation level. This field is optional. If not specified, the default reputation level image is used. Available in API version 32.0 and later.
`label`	string	Name of the reputation level. This field is optional. If not specified, one of the 10 defaults is used. • Level 1 • Level 2 • Level 3 • Level 4 • Level 5 • Level 6 • Level 7 • Level 8 • Level 9 • Level 10
`lowerThreshold`	double	Required. The lower value in the range for this reputation level. For example, if this reputation level is for points 1–50, 1 is the `lowerThreshold`.

ReputationPointsRules

Represents points rules in an Experience Cloud site's point system.

Field	Field Type	Description
pointsRule	ReputationPointsRule[]	Represents events and their associated points.

ReputationPointsRule

Represents the event and associated point value for a points rule. When a user acts, they accrue the associated points.

Field	Field Type	Description
eventType	string	Required. The type of event a member has to perform to get points. The available values are: • FeedItemWriteAPost • FeedItemWriteAComment • FeedItemReceiveAComment • FeedItemLikeSomething • FeedItemReceiveALike • FeedItemMentionSomeone • FeedItemSomeoneMentionsYou • FeedItemShareAPost • FeedItemSomeoneSharesYourPost • FeedItemPostAQuestion • FeedItemAnswerAQuestion • FeedItemReceiveAnAnswer • FeedItemMarkAnswerAsBest • FeedItemYourAnswerMarkedBest • FeedItemEndorseSomeoneForKnowledgeOnATopic • FeedItemEndorsedForKnowledgeOnATopic
points	int	Required. The number of points a member gets for performing the event. The default number of points per event is: • FeedItemWriteAPost +1 • FeedItemWriteAComment: +1 • FeedItemReceiveAComment: +5 • FeedItemLikeSomething: +1 • FeedItemReceiveALike: +5 • FeedItemMentionSomeone: +1 • FeedItemSomeoneMentionsYou: +5 • FeedItemShareAPost: +1 • FeedItemSomeoneSharesYourPost: +5 • FeedItemPostAQuestion: +1

Field	Field Type	Description
		• FeedItemAnswerAQuestion: +5 • FeedItemReceiveAnAnswer: +5 • FeedItemMarkAnswerAsBest: +5 • FeedItemYourAnswerMarkedBest: +20 • FeedItemEndorseSomeoneForKnowledgeOnATopic: +5 • FeedItemEndorsedForKnowledgeOnATopic: +20

ScheduledRecommendation

Represents a list of scheduled recommendations. Available in API version 41.0 and later.

Field	Field Type	Description
`scheduledRecommendationDetails`	ScheduledRecommendationDetail[]	A list of scheduled recommendations.

ScheduledRecommendationDetail

The specific details of a scheduled recommendation. Available in API version 41.0 and later.

Field	Field Type	Description
`channel`	RecommendationChannel (enumeration of type string)	A way to group recommendations together to determine where they show up in the site. The valid values are: • `DefaultChannel`—The default recommendation channel. Recommendations in the default channel appear in predefined locations, such as directly in the feed in Salesforce mobile web and on the Home and Question Detail pages of the Customer Service (Napili) template. • `CustomChannel1`—A custom recommendation channel. Choose where you want recommendations to appear by adding the Recommendations Carousel component to the page in Experience Builder. • `CustomChannel2`—A custom recommendation channel. • `CustomChannel3`—A custom recommendation channel.

Field	Field Type	Description
		• `CustomChannel4`—A custom recommendation channel. • `CustomChannel5`—A custom recommendation channel.
`enabled`	boolean	Indicates whether scheduling is enabled. If `true`, the recommendation is enabled and appears in sites. If `false`, recommendations in feeds in Salesforce mobile web aren't removed, but no new recommendations appear. In sites, disabled recommendations no longer appear.
`rank`	int	The rank of the recommendation within the channel, which determines the order in which it's displayed. The scheduled recommendation is inserted into the position specified by the rank. The rank of all the scheduled recommendations after it is pushed down. If the specified rank is larger than the size of the list, the scheduled recommendation is put at the end of the list. If a rank isn't specified, the scheduled recommendation is put at the end of the list.
`recommendationAudience`	string	The name of the audience for this scheduled recommendation.

NetworkTabSet

Field	Field Type	Description
`customTab`	string	Custom tab that is part of the site.
`defaultTab`	string	The Home tab for the site. When members log in, this tab is the first page they see.
`standardTab`	string	Standard tab that is part of the site.

Declarative Metadata Sample Definition

A sample XML definition of a network.

```
<?xml version="1.0" encoding="UTF-8"?>
<Network xmlns="http://soap.sforce.com/2006/04/metadata">
    <allowMembersToFlag>true</allowMembersToFlag>

<changePasswordTemplate>unfiled$public/CommunityChangePasswordEmailTemplate</changePasswordTemplate>

    <description>Metadata Community</description>
    <emailSenderAddress>admin@networkMetadata.com</emailSenderAddress>
    <emailSenderName>Admin User</emailSenderName>
    <enableInvitation>false</enableInvitation>
    <enableKnowledgeable>true</enableKnowledgeable>
    <enableNicknameDisplay>false</enableNicknameDisplay>
    <enablePrivateMessages>true</enablePrivateMessages>
    <enableReputation>true</enableReputation>
    <enableUpDownVote>true</enableUpDownVote>

<forgotPasswordTemplate>unfiled$public/CommunityForgotPasswordEmailTemplate</forgotPasswordTemplate>

    <networkMemberGroups>
        <permissionSet>Admin</permissionSet>
        <permissionSet>Standard</permissionSet>
        <permissionSet>ReadOnly</permissionSet>
        <profile>Admin</profile>
        <profile>Standard</profile>
        <profile>ReadOnly</profile>
    </networkMemberGroups>
    <recommendationDefinition>
        <recommendationDefinitionDetails>
            <actionUrl>https://www.apple.com/iphone</actionUrl>
            <description>Better specs and high performance for iPhones</description>
            <linkText>iPhone 7</linkText>
            <scheduledRecommendations>
                <scheduledRecommendationDetails>
                    <channel>DefaultChannel</channel>
                    <enabled>false</enabled>
                    <rank>1</rank>
                    <recommendationAudience>New Member Audience</recommendationAudience>
                </scheduledRecommendationDetails>
            </scheduledRecommendations>
            <setupName>Apple iPhone</setupName>
            <title>iPhone7</title>
        </recommendationDefinitionDetails>
        <recommendationDefinitionDetails>
            <actionUrl>https://www.bose.com/qc35</actionUrl>
            <description>New Amazing Noise cancellation Headphones</description>
            <linkText>Bose QC35</linkText>
            <scheduledRecommendations>
                <scheduledRecommendationDetails>
                    <channel>DefaultChannel</channel>
                    <enabled>true</enabled>
                    <rank>2</rank>
```

```
                    <recommendationAudience>Custom Audience</recommendationAudience>
                </scheduledRecommendationDetails>
            </scheduledRecommendations>
            <setupName>Bose Headphones</setupName>
            <title>Bose QC35</title>
        </recommendationDefinitionDetails>
    </recommendationDefinition>
    <reputationLevels>
        <level>
            <branding>
                <smallImage>communities_shared
_document_folder/replevel_beginner.png</smallImage>
            </branding>
            <label>Beginner</label>
            <lowerThreshold>0</lowerThreshold>
        </level>
        <level>
            <branding>
                <smallImage>communities_shared
_document_folder/replevel_apprentice.png</smallImage>
            </branding>
            <label>Apprentice</label>
            <lowerThreshold>51</lowerThreshold>
        </level>
        <level>
            <branding>
                <smallImage>communities_shared
_document_folder/replevel_gettingthere.png</smallImage>
            </branding>
            <label>Getting There</label>
            <lowerThreshold>101</lowerThreshold>
        </level>
        <level>
            <branding>
                <smallImage>communities_shared
_document_folder/replevel_skilled.png</smallImage>
            </branding>
            <label>Skilled</label>
            <lowerThreshold>151</lowerThreshold>
        </level>
        <level>
            <branding>
                <smallImage>communities_shared
_document_folder/replevel_expert.png</smallImage>
            </branding>
            <label>Expert</label>
            <lowerThreshold>201</lowerThreshold>
        </level>
        <level>
            <branding>
                <smallImage>communities_shared
_document_folder/replevel_mentor.png</smallImage>
            </branding>
            <label>Mentor</label>
```

```
            <lowerThreshold>251</lowerThreshold>
        </level>
        <level>
            <branding>
                <smallImage>communities_shared
_document_folder/replevel_guru.png</smallImage>
            </branding>
            <label>Guru</label>
            <lowerThreshold>301</lowerThreshold>
        </level>
    </reputationLevels>
    <reputationPointsRules>
        <pointsRule>
            <eventType>FeedItemWriteAPost</eventType>
            <points>5</points>
        </pointsRule>
        <pointsRule>
            <eventType>FeedItemWriteAComment</eventType>
            <points>3</points>
        </pointsRule>
        <pointsRule>
            <eventType>FeedItemReceiveAComment</eventType>
            <points>10</points>
        </pointsRule>
        <pointsRule>
            <eventType>FeedItemLikeSomething</eventType>
            <points>3</points>
        </pointsRule>
        <pointsRule>
            <eventType>FeedItemReceiveALike</eventType>
            <points>5</points>
        </pointsRule>
        <pointsRule>
            <eventType>FeedItemMentionSomeone</eventType>
            <points>5</points>
        </pointsRule>
        <pointsRule>
            <eventType>FeedItemSomeoneMentionsYou</eventType>
            <points>10</points>
        </pointsRule>
        <pointsRule>
            <eventType>FeedItemShareAPost</eventType>
            <points>5</points>
        </pointsRule>
        <pointsRule>
            <eventType>FeedItemSomeoneSharesYourPost</eventType>
            <points>10</points>
        </pointsRule>
    </reputationPointsRules>
    <selfRegistration>false</selfRegistration>
    <sendWelcomeEmail>true</sendWelcomeEmail>
    <site>Network_11</site>
    <status>UnderConstruction</status>
    <tabs>
```

```
        <defaultTab>Chatter</defaultTab>
        <standardTab>Chatter</standardTab>
        <standardTab>Account</standardTab>
        <standardTab>Campaign</standardTab>
        <standardTab>Case</standardTab>
        <standardTab>Console</standardTab>
        <standardTab>Contact</standardTab>
        <standardTab>Contract</standardTab>
        <standardTab>Dashboard</standardTab>
        <standardTab>JigsawSearch</standardTab>
        <standardTab>File</standardTab>
        <standardTab>CollaborationGroup</standardTab>
        <standardTab>home</standardTab>
        <standardTab>Idea</standardTab>
        <standardTab>Lead</standardTab>
        <standardTab>Opportunity</standardTab>
        <standardTab>Product2</standardTab>
        <standardTab>UserProfile</standardTab>
        <standardTab>report</standardTab>
        <standardTab>Solution</standardTab>
    </tabs>
    <urlPathPrefix>network1</urlPathPrefix>
    <welcomeTemplate>unfiled$public/CommunityWelcomeEmailTemplate</welcomeTemplate>
</Network>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SEE ALSO:

Community (Zone)

NetworkBranding

Represents the branding and color scheme applied to the login pages of an Experience Cloud site. (Experience Cloud sites are represented by the Network component.)

This type extends the MetadataWithContent type and inherits its `content` and `fullName` fields.

Note: For branding properties that apply to Experience Builder sites, see BrandingSet.

Declarative Metadata File Suffix and Directory Location

NetworkBranding components have the suffix `.networkBranding` and are stored in the `networkBranding` folder.

Version

This object is available in API version 41.0 and later. It replaces the Branding subtype in the Network component.

Fields

Field	Field Type	Description
`loginBackgroundImageUrl`	string	The path to the image URL that appears as the background on the Experience Cloud site's login page. This URL can be fixed, dynamic, or an uploaded image. A dynamic URL contains the experience ID parameter, `{expid}`, and is resolved dynamically at runtime.
`loginFooterText`	string	The text that appears in the footer of the Experience Cloud site login page.
`loginLogo`	string	The logo that appears on the Experience Cloud site login page for external users.
`loginLogoName`	string	The name of the logo that appears on the Experience Cloud site login page for external users.
`loginPrimaryColor`	string	The background color of the Login button. Available in API version 42.0 and later.
`loginQuaternaryColor`	string	The background color for the Experience Cloud site's login page.
`loginRightFrameUrl`	string	The path to the content of the right frame of the Experience Cloud site login page. This URL can be either fixed or dynamic. A dynamic URL contains the experience ID parameter, `{expid}`. If the URL contains `{expid}`, the URL is resolved dynamically at runtime depending on the parameter's value.
`network`	string	The name of the Experience Cloud site associated with the branding.
`pageFooter`	string	An image that appears on the footer of the Experience Cloud site pages. Must be an .html file.
`pageHeader`	string	An image that appears on the header of the Experience Cloud site pages. Can be an .html, .gif, .jpg, or .png file.
`primaryColor`	string	Required. The color used for the active tab.
`primaryComplementColor`	string	Required. Font color used with `primaryColor`.

Field	Field Type	Description
quaternaryColor	string	Required. The background color for pages in the Experience Cloud site.
quaternaryComplementColor	string	Required. Font color used with quaternaryColor.
secondaryColor	string	Required. The color used for the top borders of lists and tables.
staticLogoImageUrl	string	The path to the logo that appears on the Experience Cloud site's login page. This URL can be fixed, dynamic, or an uploaded image. A dynamic URL contains the experience ID parameter, {expid}. If the URL contains {expid}, the URL is resolved dynamically at runtime depending on the parameter's value.
tertiaryColor	string	Required. The background color for section headers on edit and detail pages.
tertiaryComplementColor	string	Required. Font color used with tertiaryColor.
zeronaryColor	string	Required. The background color for the header.
zeronaryComplementColor	string	Required. Font color used with zeronaryColor.

Declarative Metadata Sample Definition

A sample XML definition of network branding.

```
<?xml version="1.0" encoding="UTF-8"?>
<NetworkBranding xmlns="http://soap.sforce.com/2006/04/metadata">
    <loginFooterText>salesforce.com</loginFooterText>
    <loginLogo>Communities_Shared_Document_Folder/header2_png.png</loginLogo>
    <loginLogoName>header2.png</loginLogoName>

<loginBackgroundImageUrl>http://identitycms.herokuapp.com/promo-background.jpg</loginBackgroundImageUrl>

    <loginQuaternaryColor>#B1BAC1</loginQuaternaryColor>
    <loginRightFrameUrl>https://www.example.com/test</loginRightFrameUrl>
    <network>Network 1</network>
    <pageFooter>Branding/footer_html.html</pageFooter>
    <pageHeader>Branding/header_Image.jpg</pageHeader>
    <primaryColor>#AF5800</primaryColor>
    <primaryComplementColor>#FFFFFF</primaryComplementColor>
    <quaternaryColor>#286FB8</quaternaryColor>
    <quaternaryComplementColor>#FFFFFF</quaternaryComplementColor>
```

```
    <secondaryColor>#000000</secondaryColor>
    <tertiaryColor>#FFFFFF</tertiaryColor>
    <tertiaryComplementColor>#222222</tertiaryComplementColor>
    <zeronaryColor>#0A3764</zeronaryColor>
    <zeronaryComplementColor>#FFFFFF</zeronaryComplementColor>
</NetworkBranding>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

NotificationTypeConfig

Represents the metadata associated with org-level notification settings for standard and custom notification types. This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

NotificationTypeConfig components have the suffix `.config` and are stored in the `notificationTypeConfig` folder.

Version

NotificationTypeConfig components are available in API version 48.0 and later.

Fields

Field Name	Field Type	Description
`notificationTypeSettings`	NotificationTypeSettings on page 1511[]	An array of delivery settings for an org's notification types.

NotificationTypeSettings

Represents the delivery settings for a standard or custom notification type.

Field Name	Field Type	Description
`notificationType`	string	Required. Specifies a notification type's API name. For standard notification types, this is the predefined API name of the standard notification type. For custom notification types, this is the API name of the custom notification type. If a custom notification type was installed with a managed package, it includes the namespace prefix. Retrieve NotificationTypeConfig to see the API names of the notification types available in your org.

Field Name	Field Type	Description
appSettings	AppSettings on page 1512[]	An array of settings for the connected apps supported for a notification type.
notificationChannels	NotificationChannels on page 1512	Defines the delivery channels for a notification type.

AppSettings

Represents the settings for the connected apps supported for a notification type.

Field Name	Field Type	Description
connectedAppName	string	Required. Specifies the API name of a connected app. If a connected app was installed with a managed package, it includes the namespace prefix. Retrieve NotificationTypeConfig to see the API names of the connected apps supported for a notification type.
enabled	boolean	Indicates whether a connected app is enabled (`true`) or not (`false`) for the notification type.

NotificationChannels

Represents the settings for the delivery channels for a notification type.

Field Name	Field Type	Description
desktopEnabled	boolean	Indicates whether desktop notifications are enabled (`true`) or not (`false`).
mobileEnabled	boolean	Indicates whether mobile notifications are enabled (`true`) or not (`false`).
slackEnabled	boolean	Indicates whether Slack notifications are enabled (`true`) or not (`false`).

Declarative Metadata Sample Definition

The following is an example of a NotificationTypeConfig component.

```
<NotificationTypeConfig xmlns="http://soap.sforce.com/2006/04/metadata">
    <notificationTypeSettings>
        <notificationType>chatter_mention</notificationType>
        <notificationChannels>
            <desktopEnabled>false</desktopEnabled>
            <mobileEnabled>true</mobileEnabled>
        </notificationChannels>
        <appSettings>
            <connectedAppName>Datawatch</connectedAppName>
            <enabled>false</enabled>
        </appSettings>
            <appSettings>
            <connectedAppName>package2__ConnectedApp2</connectedAppName>
```

```
            <enabled>true</enabled>
        </appSettings>
    </notificationTypeSettings>
   <notificationTypeSettings>
        <notificationType>namespace__Custom_Notification</notificationType>
        <notificationChannels>
            <desktopEnabled>true</desktopEnabled>
            <mobileEnabled>true</mobileEnabled>
        </notificationChannels>
        <appSettings>
            <connectedAppName>namespace__Connected_App</connectedAppName>
            <enabled>false</enabled>
        </appSettings>
        <appSettings>
            <connectedAppName>namespace2__ConnectedApp2</connectedAppName>
            <enabled>true</enabled>
        </appSettings>
    </notificationTypeSettings>
</NotificationTypeConfig>
```

The following is an example of a package manifest used to retrieve all the available notification settings for an organization, using a wildcard:

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>NotificationTypeConfig</name>
    </types>
    <version>48.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

OauthCustomScope

Represents a permission defining the protected data that a connected app can access from an external entity when Salesforce is the OAuth authorization provider. This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

OauthCustomScope components have the suffix `.oauthcustomscope` and are stored in the `oauthcustomscopes` directory.

Version

OAuth custom scopes are available in API version 46.0 and later.

Special Access Rules

You must have the "Manage Connected Apps" permission to access this object.

Fields

Field Name	Field Type	Description
`assignedTo`	OauthCustomScopeApp (enumeration of type string)	Represents the name of the connected app to which the custom scope is assigned. Available in API version 49.0 and later. If the connected app is part of a package, include the package's namespace prefix with the connected app's name. Use the following format: ***<namespace_prefix>__<connected_app>***. Use two underscores (_) between the namespace prefix and connected app's name.
`description`	string	Required. The description of the permission provided to the connected app by the scope. The custom scope's description must be unique, can only include alphanumeric characters, and can be up to 60 characters long. You can enter a custom label in place of a description. An advantage of using a custom label is that you can maintain reusable text in a single location and translate the text into multiple languages. See Custom Labels. Note: The description formatting requirements that apply to custom scopes also apply to custom labels.
`developerName`	string	Required. Use when referring to the OAuth custom scope from a program.
`isProtected`	boolean	Required. Indicates whether this component is protected () or not (`false`). Protected components cannot be linked to or referenced by components created in the installing org.
`isPublic`	boolean	Indicates whether the object is included in the connected app's OpenID Connect discovery endpoint. The default setting is `false`. For more information, see OpenID Connect Discovery Endpoint.
`masterLabel`	string	Required. The primary label for the custom scope record. This label must be unique and begin with a letter. It can include only alphanumeric characters and underscores. It can't contain spaces.

Declarative Metadata Sample Definition

The following is an example of an OAuthCustomScope component. In this example, `basicScope` is the name of custom scope entity being retrieved.

```
<?xml version="1.0" encoding="UTF-8"?>
<OauthCustomScope xmlns="http://soap.sforce.com/2006/04/metadata">
    <assignedTo>
        <connectedApp>MyOrgNamespace__TestApp</connectedApp>
    </assignedTo>
    <description>Example of a basic custom scope</description>
    <developerName>basicScope</developerName>
    <masterLabel>basicScope</masterLabel>
    <isProtected>false</isProtected>
    <isPublic>true</isPublic>
</OauthCustomScope>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
<types>
<members>basicScope</members>
<name>OauthCustomScope</name>
</types>
<version>49.0</version>
</Package>
```

Usage

An OAuth custom scope tells an external entity about a connected app's permissions to access protected data. The OAuth custom scope you create in your Salesforce org corresponds to the same custom scope defined in your external entity and assigned to the resource.

For example, you define an Order Status custom scope in your external entity that allows access to customer order status data in your order system's API. In Salesforce, you create an OAuth custom scope that you also name Order Status. You assign this custom scope to the connected app requesting access to the order status API. When the external entity receives the connected app's request to access a customer's order status, it validates the connected app's access token and Order Status scope. With a successful validation, the app can access the customer order status information in the order system's API.

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

OauthTokenExchangeHandler

Represents a token exchange handler. The token exchange handler also consists of an Apex class. During the OAuth 2.0 token exchange flow, the token exchange handler is used to validate tokens from an external identity provider and to map users to Salesforce.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

OauthTokenExchangeHandler components have the suffix `.oauthtokenexchangehandler` and are stored in the `oauthtokenexchangehandlers` folder.

Version

OauthTokenExchangeHandler components are available in API version 60.0 and later.

Special Access Rules

There are no additional access requirements that are specific to this type.

Fields

Field Name	Description
`description`	**Field Type** string **Description** Required. A description for your token exchange handler.
`developerName`	**Field Type** string **Description** Required. The API name for the handler.
`enablements`	**Field Type** OauthTokenExchHandlerApp[] **Description** The enablement settings for the token exchange handler, including the execution user who runs the Apex class, the connected apps or external client apps for which it's enabled, and whether or not it's the default handler.
`isAccessTokenSupported`	**Field Type** boolean **Description** Required. Indicates whether the handler supports OAuth 2.0 access tokens from the identity provider, including opaque access tokens and JSON Web Token (JWT)-based access tokens.

Field Name	Description
`isEnabled`	**Field Type** boolean **Description** Required. Indicates whether the handler is enabled. To complete enablement, add an `enablements` field that specifies the enablement settings.
`isIdTokenSupported`	**Field Type** boolean **Description** Required. Indicates whether the handler supports OpenID Connect ID tokens from the identity provider.
`isJwtSupported`	**Field Type** boolean **Description** Required. Indicates whether the handler supports tokens from the identity provider that are in JWT format, such as JWT-based access tokens.
`isProtected`	**Field Type** boolean **Description** Indicates whether the handler can be linked to or referenced by components created in a subscriber org. See Protected Components in Managed Packages.
`isRefreshTokenSupported`	**Field Type** boolean **Description** Required. Indicates whether the handler supports OAuth 2.0 refresh tokens from the identity provider.
`isSaml2Supported`	**Field Type** boolean **Description** Required. Indicates whether the handler supports SAML 2.0 assertions from the identity provider.
`isUserCreationAllowed`	**Field Type** boolean **Description** Required. Indicates whether the handler can set up new users. During the token exchange flow, the Apex handler maps users from the identity provider to Salesforce. If the `isUserCreationAllowed` field is `true`, the `canCreateUser` boolean

Field Name	Description
	in the `getUserForTokenSubject` method is `true`, and the user doesn't exist in Salesforce, the handler sets up a new User object, which Salesforce automatically inserts to finish creating the user.
`masterLabel`	**Field Type** string **Description** Required. The label of the token exchange handler record.
`tokenHandlerApex`	**Field Type** string **Description** Required. The Apex class associated with the token exchange handler. The class contains methods to validate the token and map users to Salesforce. It must extend the `Oauth2TokenExchangeHandler` Apex class.

OauthTokenExchHandlerApp

Represents the settings for a specific Salesforce connected app or external client app that's enabled for the token exchange handler. A handler can be enabled for multiple apps.

Field Name	Description
`apexExecutionUser`	**Field Type** string **Description** Required. A user who runs the Apex token exchange handler. We recommend that you use an integration user.
`connectedApp`	**Field Type** string **Description** The API name of the connected app that's being used to integrate with Salesforce.
`externalClientApp`	**Field Type** string **Description** The API name of the external client app that's being used to integrate with Salesforce.
`isDefault`	**Field Type** boolean

Field Name	Description
	Description Required. Indicates whether the token exchange handler is the default handler for this app. During the token exchange flow, in the token request, you can optionally include a `token_handler` parameter with the name of a specific handler's Apex class. If you don't include this parameter, Salesforce defaults to the default handler.

Declarative Metadata Sample Definition

The following is an example of an OauthTokenExchangeHandler component.

```
<?xml version="1.0" encoding="UTF-8"?>
<OauthTokenExchangeHandler xmlns="http://soap.sforce.com/2006/04/metadata">
    <developerName>MyTokenExchangeHandler</developerName>
    <description>My token exchange handler</description>
    <isAccessTokenSupported>true</isAccessTokenSupported>
    <isEnabled>true</isEnabled>
    <isIdTokenSupported>false</isIdTokenSupported>
    <isJwtSupported>true</isJwtSupported>
    <isProtected>false</isProtected>
    <isRefreshTokenSupported>false</isRefreshTokenSupported>
    <isSaml2Supported>false</isSaml2Supported>
    <isUserCreationAllowed>true</isUserCreationAllowed>
    <masterLabel>MyTokenExchangeHandler</masterLabel>
    <tokenHandlerApex>MyOauthTokenExchangeHandler</tokenHandlerApex>
    <enablements>
        <apexExecutionUser>integrationuser@mycompany.com</apexExecutionUser>
        <connectedApp>TokenExchangeApp1</conectedApp>
        <isDefault>true</isDefault>
    </enablements>
</OauthTokenExchangeHandler>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>OauthTokenExchangeHandler</name>
    </types>
    <version>60.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

OcrSampleDocument

Represents the details of a sample document or a document type that's used as a reference while extracting and mapping information from a customer form. This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

The OcrSampleDocument type doesn't need to represent a real sample document. It can also be an abstract document that represents all documents of the same DocumentType. In such cases, the `contentAsset` and `documentHeight` fields are null.

File Suffix and Directory Location

OcrSampleDocument components have the suffix `.ocrSampleDocument` and are stored in the `ocrSampleDocuments` folder.

Version

OcrTemplate components are available in API version 52.0 and later.

Special Access Rules

To use this metadata type, your Salesforce org must have the AWSTextract1000LimitAddOn license.

Fields

Field Name	Field Type	Description
`applicationType`	OcrApplicationType (enumeration of type string)	The type of application using the OCR sample document. Possible values are: • `EinsteinDocumentReader` • `Industries`
`contentAsset`	string	The ID of the OCR sample document asset. This field is null if the OcrSampleDocument is an abstract document representing the DocumentType.
`documentHeight`	double	The normalized height of the OCR sample document page. This field is null if the OcrSampleDocument is an abstract document representing the DocumentType.
`documentType`	string	Required. The type of the OCR sample document.
`masterLabel`	string	Required. The label for the OCR sample document.
`ocrSampleDocumentFields`	OcrSampleDocumentField[]	The details of the field in a form whose value is extracted and mapped to a Salesforce object field.

Field Name	Field Type	Description
ocrSampleDocumentPages	OcrSampleDocumentPage[]	A collection of fields that define a page in the OCR sample document.

OcrSampleDocumentField

Represents the details of the field in a form whose value is extracted and mapped to a Salesforce object field.

Table 3: Fields

Field Name	Field Type	Description
cellColumnNumber	int	The column number in the item with the cell storing this field's value. Available in API version 56.0 and later.
cellColumnSpanValue	int	The number of columns that span the cell storing this field's value. Available in API version 56.0 and later.
cellRowNumber	int	The row number in the item with the cell storing this field's value. Available in API version 56.0 and later.
cellRowSpanValue	int	The number of rows that span the cell storing this field's value. Available in API version 56.0 and later.
fieldLabelMaxX	double	A normalized coordinate representing the right edge of the bounding box of the key.
fieldLabelMaxY	double	A normalized coordinate representing the bottom edge of the bounding box of the key.
fieldLabelMinX	double	A normalized coordinate representing the left edge of the bounding box of the key.
fieldLabelMinY	double	A normalized coordinate representing the top edge of the bounding box of the key.
fieldValueName	string	Name of the referred field value. Available in API version 56.0 and later.
isAutoExtractedValue	boolean	Indicates whether the key is automatically extracted (`true`) or not (`false`). Available in API version 57.0 and later. This field helps to distinguish auto-extracted keys from manual ones.
keyContent	string	The content in a particular area of the form, representing the field that is extracted by OCR.
ocrSampleDocument	string	Required. The associated OCR sample document used as a reference while extracting and mapping information from a customer form.
ocrSampleDocumentPage	string	A reference to a page of the OCR sample document that contains the key. This field is null if the OcrSampleDocument is an abstract document representing the DocumentType.
ocrSampleDocumentPageItem	OcrTemplate	A reference to the item on the sample document page containing this field's value. Available in API version 56.0 and later.

OcrSampleDocumentPage

Represents a collection of fields that define a page in the OCR sample document. This type exists only if the OcrSampleDocument is a real sample document and not an abstract document representing the DocumentType.

Table 4: Fields

Field Name	Field Type	Description
`ocrSampleDocument`	string	Required. The associated OCR sample document used as a reference while extracting and mapping information from a customer form.
`ocrSampleDocumentPageItems`	OcrSampleDocument	The collection of page items with the associated OCR sample document page. Available in API version 56.0 and later.
`pageHeight`	double	The normalized height of the OCR sample document page.
`pageNumber`	integer	Required. The page number of the page in the associated OCR sample document.

OcrSampleDocumentPageItem

Represents a foreign key reference to the item on the sample document page containing a value for the page item.

Table 5: Fields

Field Name	Field Type	Description
`hasHeader`	boolean	Indicates whether the OCR sample document page item has a header (`true`) or not (`false`). The default value is `false`. Available in API version 56.0 and later.
`sequenceNumber`	int	Required. The sequence number of the item on an OCR sample document page with multiple items. Available in API version 56.0 and later.
`title`	string	The title of the OCR sample document page item. Available in API version 56.0 and later.
`type`	ItemType (enumeration of type string)	Required. Specifies the type of OCR sample document page item. Available in API version 56.0 and later. Valid value is `TABLE`.

Declarative Metadata Sample Definition

The following is an example of a OcrSampleDocument component.

```
<?xml version="1.0" encoding="UTF-8"?>
<OcrSampleDocument xmlns="http://soap.sforce.com/2006/04/metadata">
   <contentAsset>asset_01jpeg</contentAsset>
   <documentHeight>1.24</documentHeight>
   <documentType>Form</documentType>
   <masterLabel>Form</masterLabel>
```

```
   <ocrSampleDocumentFields>
   ...<cellColumnNumber>1</cellColumnNumber>
      <cellColumnSpanValue>1</cellColumnSpanValue>
      <cellRowNumber>1</cellRowNumber>
      <cellRowSpanValue>1</cellRowSpanValue>
      <fieldLabelMaxX>0.5975854</fieldLabelMaxX>
      <fieldLabelMaxY>0.46625894</fieldLabelMaxY>
      <fieldLabelMinX>0.5065626</fieldLabelMinX>
      <fieldLabelMinY>0.39605626</fieldLabelMinY>
      <keyContent>Last Name</keyContent>
     <ocrSampleDocument>image240</ocrSampleDocument>
     <ocrSampleDocumentPage>1</ocrSampleDocumentPage>
     <ocrSampleDocumentPageItem>
         <hasHeader>false</hasHeader>
         <sequenceNumber>1</sequenceNumber>
         <title>Table1</title>
         <type>TABLE</type>
     </ocrSampleDocumentPageItem>
   </ocrSampleDocumentFields>
   <ocrSampleDocumentPages>
      <ocrSampleDocument>Form</ocrSampleDocument>
      <pageHeight>1.0</pageHeight>
      <pageNumber>1</pageNumber>
   </ocrSampleDocumentPages>
   <ocrSampleDocumentPages>
      <ocrSampleDocument>Form</ocrSampleDocument>
      <pageNumber>2</pageNumber>
   </ocrSampleDocumentPages>
</OcrSampleDocument>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
   <types>
      <members>*</members>
      <name>DocumentType</name>
   </types>
   <types>
      <members>*</members>
      <name>ContentAsset</name>
   </types>
   <types>
      <members>*</members>
      <name>OcrSampleDocument</name>
   </types>
   <version>64.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

OcrTemplate

Represents the details of the mapping between a form and a Salesforce object using Intelligent Form Reader.This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

OcrTemplate components have the suffix `.ocrTemplate` and are stored in the `ocrTemplates` folder.

Version

OcrTemplate components are available in API version 52.0 and later.

Special Access Rules

To use this metadata type, your Salesforce org must have the AWSTextract1000LimitAddOn license.

Fields

Field Name	Field Type	Description
`active`	boolean	Indicates whether the OCR template is active (`true`) or not (`false`).
`description`	string	The description of the OCR template.
`documentType`	string	Required. The document type for which this template defines mappings.
`masterLabel`	string	Required. The label for the OCR template.
`ocrTargetObjects`	OcrTargetObject[]	Represents the details of the object to which information from a form is extracted and mapped.
`ocrTemplateSampleDocuments`	OcrTemplateSampleDocument[]	Represents the details of a sample document or a document type that's used as a reference while extracting and mapping information from a customer form.
`pageCount`	integer	The number of pages in the form from which information is extracted.
`templateName`	string	Required. The name of the OCR template.

OcrTargetObject

Represents the details of the object to which information from a form is extracted and mapped.

Table 6: Fields

Field Name	Field Type	Description
ocrTargetObjectFieldMappings	OcrTargetObjFieldMapping[]	Represents the details of how information from a form field is mapped to fields in an object.
targetObject	string	Required. The object to which information from a form is mapped.
targetObjectRecordType	string	The developer name of the record type of the target object. Available in API version 56.0 and later.

OcrTargetObjFieldMapping

Represents the details of how information from a form field is mapped to fields in an object.

Table 7: Fields

Field Name	Field Type	Description
ocrSampleDocField	OcrSampleDocumentField[]	The details of the field in a form whose value is extracted and mapped to a Salesforce object field.
targetField	string	Required. The field to which information is mapped.
type	OcrMappingType (enumeration of type string)	Required. Specifies the type of mapping. Available in API version 56.0 and later. Valid values are: • FormField • TableColumn The default value is FormField.

OcrSampleDocumentField

Represents the details of the field in a form whose value is extracted and mapped to a Salesforce object field.

Table 8: Fields

Field Name	Field Type	Description
cellColumnNumber	int	The column number in the item with the cell storing this field's value. Available in API version 56.0 and later.
cellColumnSpanValue	int	The number of columns that span the cell storing this field's value. Available in API version 56.0 and later.
cellRowNumber	int	The row number in the item with the cell storing this field's value. Available in API version 56.0 and later.
cellRowSpanValue	int	The number of rows that span the cell storing this field's value. Available in API version 56.0 and later.
fieldLabelMaxX	double	A normalized coordinate representing the right edge of the bounding box of the key.

Field Name	Field Type	Description
`fieldLabelMaxY`	double	A normalized coordinate representing the bottom edge of the bounding box of the key.
`fieldLabelMinX`	double	A normalized coordinate representing the left edge of the bounding box of the key.
`fieldLabelMinY`	double	A normalized coordinate representing the top edge of the bounding box of the key.
`fieldValueName`	string	The name of the referred field value. Available in API version 56.0 and later.
`isAutoExtractedValue`	boolean	Indicates whether the key is automatically extracted (`true`) or not (`false`). Available in API version 57.0 and later. This field helps to distinguish auto-extracted keys from manual ones.
`keyContent`	string	The content in a particular area of the form, representing the field that is extracted by OCR.
`ocrSampleDocument`	string	Required. The associated OCR sample document is used as a reference while extracting and mapping information from a customer form.
`ocrSampleDocumentPage`	string	A collection of fields that define a page in the OCR sample document.
`ocrSampleDocumentPageItem`	OcrSampleDocumentPageItem	A reference to the item on the sample document page containing this field's value. Available in API version 56.0 and later.

OcrSampleDocumentPageItem

Represents a foreign key reference to the item on the sample document page containing a value for the page item.

Table 9: Fields

Field Name	Field Type	Description
`hasHeader`	boolean	Indicates whether the OCR sample document page item has a header (`true`) or not (`false`). The default value is `false`. Available in API version 56.0 and later.
`sequenceNumber`	int	Required. The sequence number of the item on an OCR sample document page with multiple items. Available in API version 56.0 and later.
`title`	string	The title of the OCR sample document page item. Available in API version 56.0 and later.
`type`	ItemType (enumeration of type string)	Required. Specifies the type of OCR sample document page item. Valid value is `TABLE`. Available in API version 56.0 and later.

OcrTemplateSampleDocument

Represents the details of a sample document or a document type that's used as a reference while extracting and mapping information from a customer form.

Table 10: Fields

Field Name	Field Type	Description
ocrSampleDocument	string	The associated OCR sample document is used as a reference while extracting and mapping information from a customer form.

Declarative Metadata Sample Definition

The following is an example of a OcrTemplate component.

```
<?xml version="1.0" encoding="UTF-8"?>
<OcrTemplate xmlns="http://soap.sforce.com/2006/04/metadata">
   <active>false</active>
   <documentType>Form</documentType>
   <masterLabel>Form Test 222</masterLabel>
   <ocrTargetObjects>
      <ocrTargetObjFieldMappings>
         <ocrSampleDocField>
            <cellColumnNumber>1</cellColumnNumber>
            <cellColumnSpanValue>1</cellColumnSpanValue>
            <cellRowNumber>1</cellRowNumber>
            <cellRowSpanValue>1</cellRowSpanValue>
             <fieldLabelMaxX>0.5975854</fieldLabelMaxX>
             <fieldLabelMaxY>0.46625894</fieldLabelMaxY>
             <fieldLabelMinX>0.5065626</fieldLabelMinX>
             <fieldLabelMinY>0.39605626</fieldLabelMinY>
             <keyContent>Last Name</keyContent>
             <ocrSampleDocument>image240</ocrSampleDocument>
             <ocrSampleDocumentPage>1</ocrSampleDocumentPage>
             <ocrSampleDocumentPageItem>
                <hasHeader>false</hasHeader>
                <sequenceNumber>1</sequenceNumber>
                <title>Table1</title>
                <type>TABLE</type>
            </ocrSampleDocumentPageItem>
        </ocrSampleDocField>
      <targetField>Account.Name</targetField>
      <type>TableColumn</type>
      </ocrTargetObjFieldMappings>
      <targetObject>Account</targetObject>
      <targetObjectRecordType>Account.X240</targetObjectRecordType>
   </ocrTargetObjects>
   <ocrTemplateSampleDocuments>
      <ocrSampleDocument>Form</ocrSampleDocument>
   </ocrTemplateSampleDocuments>
   <pageCount>10</pageCount>
```

```
    <templateName>Form Test</templateName>
</OcrTemplate>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>OcrTemplate</name>
    </types>
    <version>64.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

OmniExtTrackingDef

Represents a connection between an OmniTrackingGroup in OmniAnalytics and a third-party Analytics system such as Google Analytics.

Note: This metadata type is part of Omnistudio Standard, not Omnistudio for Vlocity.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

OmniExtTrackingDef components have the suffix `.OmniExtTrackingDef` and are stored in the `OmniExtTrackingDefs` folder.

Version

OmniExtTrackingDef components are available in API version 60.0 and later.

Special Access Rules

Using OmniAnalytics requires having an Omnistudio license and enabling OmniAnalytics in Setup.

Fields

Field Name	Description
`description`	**Field Type** string **Description** A description of the OmniExtTrackingDef.
`developerName`	**Field Type** string **Description** Required. The unique name of the OmniExtTrackingDef in the API. This name can contain only underscores and alphanumeric characters and must be unique in your organization. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. Limit: 80 characters. Note: When creating large sets of data, always specify a unique `DeveloperName` for each record. If no `DeveloperName` is specified, performance may slow while Salesforce generates one for each record. Note: Only users with View DeveloperName OR View Setup and Configuration permission can view, group, sort, and filter this field.
`isActive`	**Field Type** boolean **Description** Required. Specifies whether the OmniExtTrackingDef is active.
`masterLabel`	**Field Type** string **Description** Required. The unique master label of the OmniExtTrackingDef. This internal label doesn't get translated.
`omniExtTrackingDefKey`	**Field Type** string **Description** A UUID generated internally by Salesforce to uniquely identify an OmniExtTrackingDef record across all orgs.

Field Name	Description
`omniExtTrackingEventDefs`	**Field Type** OmniExtTrackingEventDef[] **Description** The OmniExtTrackingEventDef objects related to this OmniExtTrackingDef.
`trackingFrameworkInformation`	**Field Type** string **Description** Required. JSON data containing information about an external service, such as the API call and input parameter names.
`trackingServiceProvider`	**Field Type** ExternalTrackingVendor (enumeration of type string) **Description** Required. The third-party Analytics system to which user interaction data is sent. Values are: • `Google` • `Mixpanel`

OmniExtTrackingEventDef

Represents a format for FlexCard or OmniScript user interaction data that a third-party Analytics system such as Google Analytics can accept.

Field Name	Description
`componentType`	**Field Type** OmniAnalyticsComponentType (enumeration of type string) **Description** Required. The type of component for which user interactions are tracked. Values are: • `Flexcard` • `Omniscript`
`description`	**Field Type** string

Field Name	Description
	Description A description of the OmniExtTrackingEventDef.
`developerName`	**Field Type** string **Description** Required. The unique name of the OmniExtTrackingEventDef in the API. This name can contain only underscores and alphanumeric characters and must be unique in your organization. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. Limit: 80 characters. Note: When creating large sets of data, always specify a unique `DeveloperName` for each record. If no `DeveloperName` is specified, performance may slow while Salesforce generates one for each record. Note: Only users with View DeveloperName OR View Setup and Configuration permission can view, group, sort, and filter this field.
`inclusionRule`	**Field Type** string **Description** Required. A true-or-false condition that determines whether an event is sent to the third-party Analytics system.
`masterLabel`	**Field Type** string **Description** Required. The unique master label of the OmniExtTrackingEventDef. This internal label doesn't get translated.
`omniExtTrackingDef`	**Field Type** string **Description** The ID of the related OmniExtTrackingDef object.
`omniExtTrackingEventDefKey`	**Field Type** string **Description** A UUID generated internally by Salesforce to uniquely identify an OmniExtTrackingEventDef record across all orgs.

Field Name	Description
payloadTemplate	**Field Type** string **Description** Required. The payload template structure with placeholders for runtime data. This is used at runtime to generate the actual payload to be sent to the external Analytics service.

Declarative Metadata Sample Definition

The following is an example of an OmniExtTrackingDef component.

```
<?xml version="1.0" encoding="UTF-8"?>
<OmniExtTrackingDef xmlns="http://soap.sforce.com/2006/04/metadata">
    <developerName>Purchase_Tracking_Google</developerName>
    <isActive>true</isActive>
    <masterLabel>Purchase_Tracking_Google</masterLabel>
    <trackingFrameworkInformation>{ "id": "GTM-XXXXXXX" }</trackingFrameworkInformation>
    <trackingServiceProvider>Google</trackingServiceProvider>
    <omniExtTrackingEventDefs>
        <componentType>Omniscript</componentType>
        <developerName>Purchase_Funnel_Google</developerName>
        <inclusionRule></inclusionRule>
        <masterLabel>Purchase_Funnel_Google</masterLabel>
        <payloadTemplate>
{
  "event": "promotionClick",
  "ecommerce": {
    "promoClick": {
      "promotions": [
        {
          "name": "%BusinessEvent%"
        }
      ]
    }
  }
}
        </payloadTemplate>
    </omniExtTrackingEventDefs>
</OmniExtTrackingDef>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>OmniExtTrackingDef</name>
    </types>
```

```
    <version>60.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

OmniInteractionAccessConfig

Represents configuration settings for access to Omnistudio FlexCard caching and data sources.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

OmniInteractionAccessConfig components have the suffix `.omniInteractionAccessConfig` and are stored in the `OmniInteractionAccessConfig` folder.

Version

OmniInteractionAccessConfig components are available in API version 53.0 and later.

Special Access Rules

OmniInteractionAccessConfig is available if your org has the Omnistudio platform license and related addon and user licenses.

Fields

Field Name	Field Type	Description
`configName`	string	Not used.
`isAsyncCardCachingEnabled`	boolean	Required. If set to `true`, enables asynchronous FlexCard caching. The default is `false`.
`isCardApexRemoteDisabled`	boolean	Required. If set to `true`, disables remote Apex method calls for FlexCards. The default is `false`.
`isCardCacheDisabled`	boolean	Required. If set to `true`, disables FlexCard caching. The default is `false`.
`isCardDataTfrmDisabled`	boolean	Required. If set to `true`, disables Data Mapper data sources for FlexCards. The default is `false`.

Field Name	Field Type	Description
`isCardIntegrationProcDisabled`	boolean	Required. If set to `true`, disables Integration Procedure data sources for FlexCards. The default is `false`.
`isCardRestApiDisabled`	boolean	Required. If set to `true`, disables REST calls for FlexCards. The default is `false`.
`isCardSoqlDisabled`	boolean	Required. If set to `true`, disables SOQL queries for FlexCards. The default is `false`.
`isCardSoslDisabled`	boolean	Required. If set to `true`, disables SOSL queries for FlexCards. The default is `false`.
`isCardStreamingApiDisabled`	boolean	Required. If set to `true`, disables Streaming API calls for FlexCards. The default is `false`.
`isDataTfrmEncrpFieldsDisabled`	boolean	Required. If set to `true`, disables Data Mapper field encryption for FlexCards. The default is `false`.
`masterLabel`	string	Required. The name of the setting. The value is `Profile_`*`ProfileId`*, `User_`*`UserId`*, or `Org_Wide`.
`setupOwner`	string	The ID of the profile, user, or org to which the settings apply.

Declarative Metadata Sample Definition

The following is an example of an OmniInteractionAccessConfig component.

```
<?xml version="1.0" encoding="UTF-8"?>
<OmniInteractionAccessConfig xmlns="http://soap.sforce.com/2021/10/metadata">
   <isAsyncCardCachingEnabled>false</isAsyncCardCachingEnabled>
   <isCardApexRemoteDisabled>false</isCardApexRemoteDisabled>
   <isCardCacheDisabled>false</isCardCacheDisabled>
   <isCardDataTfrmDisabled>false</isCardDataTfrmDisabled>
   <isCardIntegrationProcDisabled>false</isCardIntegrationProcDisabled>
   <isCardRestApiDisabled>false</isCardRestApiDisabled>
   <isCardSoqlDisabled>false</isCardSoqlDisabled>
   <isCardSoslDisabled>false</isCardSoslDisabled>
   <isCardStreamingApiDisabled>false</isCardStreamingApiDisabled>
   <isDataTfrmEncrpFieldsDisabled>false</isDataTfrmEncrpFieldsDisabled>
   <masterLabel>Profile_00eB0000000ijOH</masterLabel>
   <setupOwner>00eB0000000ijOH</setupOwner>
</OmniInteractionAccessConfig>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2021/10/metadata">
    <types>
        <members>*</members>
        <name>OmniInteractionAccessConfig</name>
    </types>
    <version>53.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

OmniInteractionConfig

Represents configuration settings for Omnistudio.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

OmniInteractionConfig components have the suffix `.omniInteractionConfig` and are stored in the `OmniInteractionConfig` folder.

Version

OmniInteractionConfig components are available in API version 51.0 and later.

Special Access Rules

OmniInteractionConfig is available if your org has the Omnistudio platform license and related add-on and user licenses.

Fields

Field Name	Field Type	Description
`masterLabel`	string	Required. The name of the setting.
`value`	string	Required. The value of the setting.

Declarative Metadata Sample Definition

The following is an example of an OmniInteractionConfig component.

```
<?xml version="1.0" encoding="UTF-8"?>
<OmniInteractionConfig xmlns="http://soap.sforce.com/2021/10/metadata">
   <masterLabel>TheFirstInstalledOmniPackage</masterLabel>
   <value>omnistudio</value>
</OmniInteractionConfig>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2021/10/metadata">
```

```
    <types>
        <members>*</members>
        <name>OmniInteractionConfig</name>
    </types>
    <version>51.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

Usage

Settings configured using OmniInteractionConfig include:

- `ApexClassCheck`—Enable access to the `VlocityOpenInterface` Apex class that's used by remote action APIs for a user profile. Configure an Apex class permissions checker to make sure that users require explicit access to the Apex class that administers the remote action called from an Omniscript, Flexcard, or REST API. Checks Apex classes assigned to a profile and controls access. If the Omnistudio license isn't available in your Salesforce org (for example, if you have installed Omnistudio as part of the CME or Insurance managed package), set the `ApexClassCheck` configuration to `true`.
- `CheckCachedMetadataRecordSecurity`—If set to `true`, performs a record-level security check for cached data in Data Mappers and Integration Procedures.
- `DefaultRequiredPermission`—The Custom Permission a user must have to run Data Mappers and Integration Procedures.
- `DocuSignAccountId`—The API Account ID from DocuSign's Apps and Keys page.
- `DocuSignNamedCredential`—The named credential for connecting to DocuSign. Set the value to `DocuSign`.
- `EnableQueryWithFLS`—Construct text-based search queries against the search index with Salesforce Object Search Language (SOSL). By using a single query, you can search text, email, and phone fields for multiple objects to which you have access, including custom objects. SOSL queries are encrypted so the query information isn't visible on the client side. To enforce field-level security for an SOSL query, add the `EnableQueryWithFLS` Omni Interaction Configuration and set it to `true`.
- `EnforceDMFLSAndDataEncryption`—If set to `true`, this setting enforces field-level security for all Data Mappers and displays encrypted fields in plain text only for users with the View Encrypted Data permission. This configuration also enforces object-level security. If set to true, Data Mappers run in the user context instead of the system context. The system respects the object and field-level permissions of the running user so that they only see the data they have access to.
- `InstalledIndustryPackage`—If present, lists the namespace of the Salesforce Industries managed package that was installed. Values are `vlocity_cmt`, `vlocity_ins`, or `vlocity_ps`. Read-only.
- `newportZipUrl`—The relative URL (without the hostname) for the static resource that contains custom Newport styles for FlexCards.
- `OmniAnalyticsTrackingDebug`—If set to `true`, includes debugging data in Omnistudio Tracking Service records.
- `PerformAdvancedCheck`—If set to `true`, this configuration enables specific access to the `VlocityOpenInterface` Apex class used by remote action APIs for a user's permission set or permission set group. This check is dependent on the `ApexClassCheck` and can only work if the `ApexClassCheck` setting is also set to `true`.
- `RetainDesignerSettingOnUpgrade`—If you're using the Standard Runtime and Package Designer, and you upgrade the package installation in your org, the designer switches to Standard Designer by default. To retain your preferred designer after the upgrade, set the `RetainDesignerSettingOnUpgrade` configuration to `true`.

Note: If you have enabled the Omnistudio standard designer by turning the Managed Package Designer setting off, you don't require this configuration. This is only applicable to orgs that use the managed package designer with the standard runtime.

- `RollbackDRChanges`—If set to `true`, rolls back Data Mapper functionality changes. Use it if an upgrade causes some Data Mappers to stop working.
- `SkipUserProfileOnOSLoad`—If set to `true`, it doesn't retrieve the user profile information when the Omniscript loads. If set to `false` or if the configuration is not available, it retrieves the user profile information when the Omniscript loads.
- `TheFirstInstalledOmniPackage`—Lists the namespace of the managed package that was installed first, which determines whether new or legacy Omnistudio features are available. Values are `omnistudio` for new features, or `vlocity_cmt`, `vlocity_ins`, or `vlocity_ps` for legacy features. Read-only.
- `Track_component`—If set to `true`, enables tracking for a component or component type in the Omnistudio Tracking Service.
- `TurnOffScaleCache`—If set to `true`, turns off the Scale Cache that Data Mappers and Integration Procedures use.

OmniScript

Represents an OmniScript for the Discovery Framework, which guides users through sales, service, and other business processes. For Discovery Framework, the customization type is *`discoveryframework`*.

This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

OmniScript components have the suffix `omniScript` and are stored in the `omniScripts` folder.

Version

OmniScript components are available in API version 56.0 and later.

Special Access Rules

To use this metadata type, you must have an Omnistudio license and the Discovery Framework feature enabled in your Salesforce org.

Fields

Field Name	Description
`customHtmlTemplates`	**Field Type** string **Description** The angular OmniScript template definitions.
`customJavaScript`	**Field Type** string

Field Name	Description
	Description The custom JavaScript used for an OmniScript.
`designerCustomizationType`	**Field Type** string **Description** The Omnistudio designer customization type.
`elementTypeComponentMapping`	**Field Type** string **Description** Overrides all elements of one type with a custom Lightning web component by mapping the element type to the Custom LWC.
`isActive`	**Field Type** boolean **Description** Indicates whether the OmniScript is active (`true`) or not (`false`). The default value is `false`.
`isIntegrationProcedure`	**Field Type** boolean **Description** Indicates whether OmniScript is an Integration Procedure (`true`) or OmniScript metadata (`false`). The default value is `false`.
`isMetadataCacheDisabled`	**Field Type** boolean **Description** Indicates whether metadata cache for the integration procedure is disabled (`true`) or not (`false`). The default value is `false`.
`isOmniScriptEmbeddable`	**Field Type** boolean **Description** Indicates whether the OmniScript can be embedded in other OmniScripts (`true`) or not (`false`). The default value is `false`
`isTestProcedure`	**Field Type** boolean **Description** Indicates whether OmniScript is a test procedure setting (`true`) or not (`false`). The default value is `false`

Field Name	Description
`isWebCompEnabled`	**Field Type** boolean **Description** Indicates whether web component OmniScript (not Angular) is enabled (`true`) or not (`false`). The default value is `false`
`language`	**Field Type** string **Description** Required. The language of the OmniScript.
`lastPreviewPage`	**Field Type** string **Description** The last page previewed in the OmniScript designer.
`name`	**Field Type** string **Description** Required. The name of the OmniScript.
`namespace`	**Field Type** string **Description** The namespace associated with this OmniScript record.
`omniAssessmentTasks`	**Field Type** OmniAssessmentTask[] **Description** The omniAssessmentTasks associated with the OmniScript. Available in API version 63.0 and later. This field is available only if the Dynamic Assessment Access license is enabled.
`omniProcessElements`	**Field Type** OmniProcessElement[] **Description** The OmniProcessElements associated with the OmniScript.
`omniProcessKey`	**Field Type** string

Field Name	Description
	Description The integration procedure `Type_SubType` value.
`omniProcessType`	**Field Type** OmniProcessType (enumeration of type string) **Description** Required. Integration Procedure or OmniScript. Possible value is: • `OmniScript`
`overrideKey`	**Field Type** string **Description** Reserved for future use.
`propertySetConfig`	**Field Type** string **Description** The configuration information associated with the OmniScript.
`questionDevName`	**Field Type** string **Description** The developer name of the assessment question in the OmniScript. Note: This field is relevant only for OmniScripts with `designerCustomizationType` as *`discoveryframework`*.
`requiredPermission`	**Field Type** string **Description** The required permissions to execute the integration procedure.
`responseCacheType`	**Field Type** string **Description** Response cache used for the integration procedure (session or Org).
`subType`	**Field Type** string **Description** Required.

Field Name	Description
	The OmniScript sub type value.
`type`	**Field Type** string **Description** Required. The OmniScript type value.
`uniqueName`	**Field Type** string **Description** Required. The unique name for the OmniScript as Type_SubType_Language_VersionNumber.
`versionNumber`	**Field Type** string **Description** Required. The OmniScript version number.
`webComponentKey`	**Field Type** string **Description** Internal unique key for the generated Lightning Web Components (LWC).

OmniAssessmentTask

Represents the omni assessment tasks associated with the OmniScript.

Field Name	Description
`name`	**Field Type** string **Description** Required. The name of the omni assessment task.
`status`	**Field Type** picklist **Description** The status of the omni assessment task.

Field Name	Description
	Possible values are: • `Completed` • `InProgress`—In Progress • `IsDefined`—Is Defined • `NotStarted`—Not Started The default value is `IsDefined`.
`uniqueName`	**Field Type** string **Description** A unique name for the omni assessment task.

OmniProcessElement

Represents the OmniScript element associated with the OmniScript.

Field Name	Description
`childElements`	**Field Type** OmniProcessElement[] **Description** The child elements associated with the OmniProcessElement.
`description`	**Field Type** string **Description** The description of the OmniProcessElement.
`designerCustomizationType`	**Field Type** string **Description** The Omnistudio designer customization type. Note: To create assessment questions using the Discovery Framework feature, use the customization type as *`discoveryframework`*.
`discoveryFrameworkUsageType`	**Field Type** string **Description** The usage type for industries that use the Discovery Framework. For example, the value for Health Cloud is `HcUsageType`. The value for no specific industry is `Default`.

Field Name	Description
`embeddedOmniScriptKey`	**Field Type** string **Description** The ID of the embedded OmniScript
`isActive`	**Field Type** boolean **Description** Indicates whether the status of the OmniProcessElement is active (`true`) or not (`false`).
`isOmniScriptEmbeddable`	**Field Type** boolean **Description** Indicates whether the OmniScript with the OmniProcessElement can be embedded in other OmniScripts (`true`) or not (`false`).
`level`	**Field Type** double **Description** The vertical level in which the OmniProcessElement occurs on the OmniScript.
`name`	**Field Type** string **Description** Required. The name of the OmniProcessElement.
`omniProcessVersionNumber`	**Field Type** double **Description** The related OmniProcess version.
`parentElementName`	**Field Type** string **Description** The name of the parent OmniProcessElement.
`parentElementType`	**Field Type** string **Description** The type of the parent OmniProcessElement.

Field Name	Description
`propertySetConfig`	**Field Type** textarea **Description** The property set of the OmniProcessElement.
`QuestionDevName`	**Field Type** string **Description** The developer name of the assessment question in the OmniScript. Note: This field is relevant only for OmniScripts with `designerCustomizationType` as *`discoveryframework`*.
`sequenceNumber`	**Field Type** double **Description** The horizontal level in which the OmniProcessElement occurs on the OmniScript.
`type`	**Field Type** string **Description** The OmniProcessElement type. For example, `Text` and `TextArea`.
`uniqueIndex`	**Field Type** string **Description** A unique index number for the OmniScript.

Declarative Metadata Sample Definition

The following is an example of a OmniScript component.

```
<?xml version="1.0" encoding="UTF-8"?>
<OmniScript
 xmlns="http://soap.sforce.com/2006/04/metadata">

<elementTypeComponentMapping>{&quot;ElementTypeToHTMLTemplateList&quot;:[]}</elementTypeComponentMapping>

 <isActive>false</isActive>
 <isIntegrationProcedure>false</isIntegrationProcedure>
 <isMetadataCacheDisabled>false</isMetadataCacheDisabled>
 <isOmniScriptEmbeddable>false</isOmniScriptEmbeddable>
 <isTestProcedure>false</isTestProcedure>
 <isWebCompEnabled>true</isWebCompEnabled>
 <language>English</language>
```

```
 <name>OS5</name>
 <omniProcessElements>
  <parentElementType></parentElementType>
  <parentElementName></parentElementName>
  <description>test</description>
  <isActive>true</isActive>
  <isOmniScriptEmbeddable>false</isOmniScriptEmbeddable>
  <embeddedOmniScriptKey>key</embeddedOmniScriptKey>
  <designerCustomizationType></designerCustomizationType>
  <level>0.0</level>
  <name>Step1</name>
  <omniProcessVersionNumber>0.0</omniProcessVersionNumber>
  <propertySetConfig>{&quot;label&quot;:&quot;Step
[illegible]
 you sure?&quot;,&quot;saveLabel&quot;:&quot;Save for
later&quot;,&quot;saveMessage&quot;:&quot;Are you sure you want to save it for
later?&quot;,&quot;completeLabel&quot;:&quot;Complete&quot;,&quot;completeMessage&quot;:&quot;Are
 you sure you want to complete the
[illegible]
 if
[illegible]

  <sequenceNumber>0.0</sequenceNumber>
  <type>Step</type>
  <questionDevName></questionDevName>
 </omniProcessElements>
 <omniProcessElements>
  <isActive>true</isActive>
  <level>0.0</level>
  <name>Step2</name>
  <omniProcessVersionNumber>0.0</omniProcessVersionNumber>
  <isOmniScriptEmbeddable>false</isOmniScriptEmbeddable>

[illegible]
 you sure?&quot;,&quot;saveLabel&quot;:&quot;Save for
later&quot;,&quot;saveMessage&quot;:&quot;Are you sure you want to save it for
later?&quot;,&quot;completeLabel&quot;:&quot;Complete&quot;,&quot;completeMessage&quot;:&quot;Are
 you sure you want to complete the
script?&quot;,&quot;instruction&quot;:&quot;&quot;,&quot;showPersistentComponent&quot;:[true,

[illegible]

&quot;keyword&quot;:&quot;&quot;,&quot;dataCategoryCriteria&quot;:&quot;&quot;,&quot;endTime&quot;:300,&quot;type&quot;:&quot;&quot;},&quot;sv&quot;:null,&quot;conditions&quot;:&quot;id
 if
[illegible]

  <sequenceNumber>1.0</sequenceNumber>
  <type>Step</type>
  <questionDevName></questionDevName>
 </omniProcessElements>
 <omniProcessElements>
  <isActive>true</isActive>
  <level>0.0</level>
  <name>Step3</name>
```

```
  <omniProcessVersionNumber>0.0</omniProcessVersionNumber>
  <isOmniScriptEmbeddable>false</isOmniScriptEmbeddable>

<propertySetConfig>{&quot;allowSaveForLater&quot;:true,&quot;autoFocus&quot;:false,&quot;cancelType&quot;:&quot;Omniscript&quot;,&quot;cancelURL&quot;:&quot;&quot;,&quot;chartOptions&quot;:{},&quot;debugMode&quot;:false,&quot;duplicateTemplateName&quot;:&quot;vlcDuplicate&quot;,&quot;duplicateTemplateType&quot;:&quot;vlocity_cmt&quot;,&quot;errorMessage&quot;:{},&quot;knowledgeOptions&quot;:{&quot;articleTypes&quot;:[]},&quot;keepHistory&quot;:false,&quot;message&quot;:{},&quot;newTemplateName&quot;:&quot;vlcNew&quot;,&quot;newTemplateType&quot;:&quot;vlocity_cmt&quot;,&quot;numberOfRevisions&quot;:10,&quot;omniScriptId&quot;:&quot;&quot;,&quot;pubsub&quot;:false,&quot;resetLabel&quot;:&quot;Reset&quot;,&quot;resetMessage&quot;:&quot;Are
 you sure?&quot;,&quot;saveLabel&quot;:&quot;Save for
later&quot;,&quot;saveMessage&quot;:&quot;Are you sure you want to save it for
later?&quot;,&quot;completeLabel&quot;:&quot;Complete&quot;,&quot;completeMessage&quot;:&quot;Are
 you sure you want to complete the
script?&quot;,&quot;instruction&quot;:&quot;&quot;,&quot;showPersistentComponent&quot;:[true,

false],&quot;stepChartType&quot;:&quot;&quot;,&quot;timeout&quot;:300,&quot;trackingCustomData&quot;:{},&quot;knowledgeOptions&quot;:{&quot;language&quot;:&quot;English&quot;,&quot;publishStatus&quot;:&quot;Online&quot;,

&quot;keyword&quot;:&quot;&quot;,&quot;dataCategoryCriteria&quot;:&quot;&quot;,&quot;remoteTimeout&quot;:3000,&quot;type&quot;:&quot;&quot;},&quot;show&quot;:null,&quot;conditionType&quot;:&quot;Hide
 if
false&quot;,&quot;message&quot;:{},&quot;sendJSONPath&quot;:&quot;&quot;,&quot;sendJSONNode&quot;:&quot;&quot;,&quot;responseJSONPath&quot;:&quot;&quot;,&quot;responseJSONNode&quot;:&quot;&quot;,&quot;remoteOptions&quot;:{},&quot;remoteTimeout&quot;:30000,&quot;elementTypeComponentMapping&quot;:{}}</propertySetConfig>

  <sequenceNumber>2.0</sequenceNumber>
  <type>Step</type>
  <questionDevName></questionDevName>
 </omniProcessElements>
 <omniProcessType>OmniScript</omniProcessType>

<propertySetConfig>{&quot;name&quot;:&quot;Type5_SBT5_English_1&quot;,&quot;bEtype&quot;:&quot;Type5&quot;,&quot;bSubType&quot;:&quot;SBT5&quot;,&quot;bLang&quot;:&quot;English&quot;,&quot;isActive&quot;:true,&quot;webComponentKey&quot;:&quot;WebComponentKey&quot;,&quot;overrideKey&quot;:&quot;&quot;,&quot;isReusable&quot;:false,&quot;isLwcEnabled&quot;:false,&quot;isSaveURLEnabled&quot;:false,&quot;isResumeEnabled&quot;:false,&quot;isSaveForLaterEnabled&quot;:false,&quot;isPreviewEnabled&quot;:false,&quot;isSingleColumnEnabled&quot;:false,&quot;isSkipConsecutiveActionSteps&quot;:false,&quot;isSkipValidationOnSave&quot;:false,&quot;isSkipValidationOnStepNavigation&quot;:false,&quot;isSkipValidationOnPreview&quot;:false,&quot;isSkipValidationOnActivation&quot;:false,&quot;isSkipValidationOnSaveForLater&quot;:false,&quot;isSkipValidationOnResume&quot;:false,&quot;isSkipValidationOnComplete&quot;:false,&quot;isSaveForLaterEnabled&quot;:false,&quot;isSaveURLEnabled&quot;:false,&quot;isResume
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

Usage Type

Before you retrieve or deploy Discovery Framework OmniScripts, we recommend that you review these considerations.

- If the `DesignerCustomizationType` of the OmniScript is `discoveryframework`, then the questions in the OmniScript must be within the *<uniqueIndex>* tag in the metadata definition file.
- When deploying the OmniScript of type Discovery Framework, enable Discovery Framework Metadata Enabled setting.
- OmniScripts of type Discovery Framework don't support IDX Workbench.
- If any question associated with the OmniScript doesn't exist in the target org or, if the active version of that question doesn't exist in the target org, then deploying the OmniScript fails.

OmniSupervisorConfig

Represents the Omni-Channel supervisor configuration for an assigned group of supervisors.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

`OmniSupervisorConfig` components have the suffix `.omniSupervisorConfig` and are stored in the `omniSupervisorConfigs` folder.

Version

OmniSupervisorConfig components are available in API version 57.0 and later.

Special Access Rules

To access this type, Omni-Channel must be enabled.

Fields

Field Name	Description
`isTimelineHidden`	**Field Type** boolean

Field Name	Description
	Description Required. If set to `true`, hides the agent timeline from the supervisors assigned to this supervisor configuration. The default value is `false`.
`masterLabel`	**Field Type** string **Description** Required. A unique label for the supervisor configuration. The name must begin with a letter and can contain alphanumeric characters and underscores. It can't contain spaces, two consecutive underscores, or end with an underscore. The name appears as Omni Supervisor Configuration Name in the UI.
`omniSupervisorConfigAction`	**Field Type** OmniSupervisorConfigAction[] **Description** Represents the actions available to the supervisors of this Omni-Channel supervisor configuration.
`omniSupervisorConfigGroup`	**Field Type** OmniSupervisorConfigGroup[] **Description** Represents the group of agents who are visible to the supervisors of an Omni-Channel supervisor configuration. The group, if visible, appears in the Agents tab of Omni Supervisor.
`omniSupervisorConfigProfile`	**Field Type** OmniSupervisorConfigProfile[] **Description** Represents the supervisor profiles to which an Omni-Channel supervisor configuration applies. User-level configurations override profile-level configurations.
`omniSupervisorConfigQueue`	**Field Type** OmniSupervisorConfigQueue[] **Description** Represents the queues that are visible to the supervisors of an Omni-Channel supervisor configuration. The queue, if visible, appears in the Queues Backlog and Assigned Work tabs of Omni Supervisor.
`omniSupervisorConfigSkill`	**Field Type** OmniSupervisorConfigSkill[] **Description** Represents the skills that are visible to the supervisors of an Omni-Channel supervisor configuration. These skills, if visible, appear in the Skills Backlog tab of Omni Supervisor.

Field Name	Description
`omniSupervisorConfigTab`	**Field Type** OmniSupervisorConfigTab[] **Description** Represents the visible tabs specified in an Omni Supervisor configuration. This object is available in API version 60.0 and later.
`omniSupervisorConfigUser`	**Field Type** OmniSupervisorConfigUser[] **Description** Represents the supervisor to which the Omni-Channel supervisor configuration applies. User-level configurations override profile-level configurations. This object is available in API version 64.0 and later.
`skillVisibility`	**Field Type** OmniSuperSkillVisibilityType (enumeration of type string) **Description** Determines which work items based on skills are visible to the supervisors assigned to this supervisor configuration. Values are: • `AllSkills` • `AnySkill`

OmniSupervisorConfigAction

Represents an action and associated tab, custom flow, and display order for supervisor actions in this configuration.

Field Name	Description
`actionName`	**Field Type** OmniSupervisorActionName (enumeration of type string) **Description** Required. An action that a supervisor can perform. Values are: • `AWSDashboard` • `AssignLearning` • `ChangeGroups` • `ChangeQueues` • `ChangeSkills` • `CustomAction` • `ManageQueues`
`actionTab`	**Field Type** OmniSupervisorActionTab (enumeration of type string)

Field Name	Description
	Description Required. The tab where the action appears. Values are: • `AgentDetails` • `AllAgents` • `AssignedWork` • `AssignedWorkDetails` • `QueueDetails` • `QueuesBacklog` • `SkillDetails` • `SkillsBacklog`
`customActionFlow`	**Field Type** string **Description** A flow that performs a custom action.
`displayOrder`	**Field Type** int **Description** Required. The order in which the action is displayed.

OmniSupervisorConfigGroup

Represents the group of agents for the supervisors in this configuration.

Field Name	Description
`group`	**Field Type** string **Description** Required. The name of the group of agents.

OmniSupervisorConfigProfile

Represents a profile for the supervisors in this configuration.

Field Name	Description
`profile`	**Field Type** string

Field Name	Description
	Description Required. The profile name.

OmniSupervisorConfigQueue

Represents a queue that's visible to the supervisors in this configuration.

Field Name	Description
`queue`	**Field Type** string **Description** Required. The queue name.

OmniSupervisorConfigSkill

Represents a skill that's visible to supervisors in this configuration.

Field Name	Description
`skill`	**Field Type** string **Description** Required. The skill name.

OmniSupervisorConfigTab

Represents the visible tabs specified in an Omni Supervisor configuration. This object is available in API version 60.0 and later.

Field	Details
`displayOrder`	**Type** int **Properties** Create, Filter, Group, Nillable, Sort **Description** The order in which tabs are displayed in Omni Supervisor.
`flexipage`	**Type** string

Field	Details
	Description The name of the FlexiPage added as a customized tab. Required when `tabType`is set to `FlexipageType`.
`tabType`	**Type** OmniSupervisorTabType (enumeration of type string) **Properties** Create, Filter, Group, Nillable, Restricted picklist, Sort **Description** Required. Tabs shown on the Omni Supervisor page. Possible values are: • `Agents` — the Agents tab • `AssignedWork` — the Assigned Work tab • `FlexipageType` — A custom tab created using Lightning App Builder, with the `omniSupervisorPageType` value of the `FlexiPage Type` field • `QueuesBacklog` — the Queues Backlog tab • `SkillsBacklog` — the Skills Backlog tab • `Wallboard` — the Wallboard tab

OmniSupervisorConfigUser

Represents the supervisor to which the Omni-Channel supervisor configuration applies. User-level configurations override profile-level configurations. This object is available in API version 64.0 and later.

Field	Details
`user`	**Type** string **Description** Required. Represents the supervisor to which the Omni-Channel supervisor configuration applies.

Declarative Metadata Sample Definition

The following is an example of a OmniSupervisorConfig component.

```
<?xml version="1.0" encoding="UTF-8"?>
<OmniSupervisorConfig xmlns="http://soap.sforce.com/2006/04/metadata">
    <isTimelineHidden>true</isTimelineHidden>
    <masterLabel>Supervisor Config</masterLabel>
    <skillVisibility>AllSkills</skillVisibility>
    <omniSupervisorConfigAction>
        <actionName>ChangeQueues</actionName>
        <actionTab>AllAgents</actionTab>
```

```
        <displayOrder>0</displayOrder>
    </omniSupervisorConfigAction>
    <omniSupervisorConfigGroup>
        <group>Agent_Group</group>
    </omniSupervisorConfigGroup>
    <omniSupervisorConfigProfile>
        <profile>contractmanager</profile>
    </omniSupervisorConfigProfile>
    <omniSupervisorConfigQueue>
        <queue>CaseQueue</queue>
    </omniSupervisorConfigQueue>
    <omniSupervisorConfigSkill>
        <skill>English</skill>
    </omniSupervisorConfigSkill>
</OmniSupervisorConfig>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>OmniSupervisorConfig</name>
    </types>
    <version>57.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

OmniTrackingGroup

Represents a group of FlexCard and OmniScript components that have their user interactions tracked together in OmniAnalytics.

Note: This metadata type is part of Omnistudio Standard, not Omnistudio for Vlocity.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

OmniTrackingGroup components have the suffix `.OmniTrackingGroup` and are stored in the `OmniTrackingGroups` folder.

Version

OmniTrackingGroup components are available in API version 60.0 and later.

Special Access Rules

Using OmniAnalytics requires having an Omnistudio license and enabling OmniAnalytics in Setup.

Fields

Field Name	Description
`description`	**Field Type** string **Description** A description of the OmniTrackingGroup.
`developerName`	**Field Type** string **Description** Required. The unique name of the OmniTrackingGroup in the API. This name can contain only underscores and alphanumeric characters and must be unique in your organization. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. Limit: 80 characters. Note: When creating large sets of data, always specify a unique `DeveloperName` for each record. If no `DeveloperName` is specified, performance may slow while Salesforce generates one for each record. Note: Only users with View DeveloperName OR View Setup and Configuration permission can view, group, sort, and filter this field.
`endDate`	**Field Type** date **Description** The date when the OmniTrackingGroup became inactive.
`groupType`	**Field Type** OmniTrackingGroupType (enumeration of type string) **Description** Required. Specifies whether this OmniTrackingGroup sends tracking data to a third-party Analytics system such as Google Analytics. Values are:

Field Name	Description
	• `External`—A third-party Analytics system is used. • `Internal`—No third-party Analytics system is used.
`isActive`	**Field Type** boolean **Description** Required. Specifies whether the OmniTrackingGroup is active.
`masterLabel`	**Field Type** string **Description** Required. The unique master label of the OmniTrackingGroup. This internal label doesn't get translated.
`maxAgeInDays`	**Field Type** int **Description** The maximum number of days the group and its analytics data is active beyond which the data is deleted.
`omniExtTrackingDef`	**Field Type** string **Description** The ID of the related OmniExtTrackingDef object. Required if `GroupType` is set to `External`.
`omniTrackingComponentDefs`	**Field Type** OmniTrackingComponentDef[] **Description** The OmniTrackingComponentDef objects related to this OmniTrackingGroup.
`omniTrackingGroupKey`	**Field Type** string **Description** A UUID generated internally by Salesforce to uniquely identify an OmniTrackingGroup record across all orgs.
`startDate`	**Field Type** date

Field Name	Description
	Description The date when the OmniTrackingGroup became active.

OmniTrackingComponentDef

Represents a FlexCard or OmniScript that is a member of an OmniTrackingGroup, which tracks user interactions in OmniAnalytics.

Field Name	Description
`componentType`	**Field Type** OmniAnalyticsComponentType (enumeration of type string) **Description** Required. The type of component for which user interactions are tracked. Values are: • `Flexcard` • `Omniscript`
`componentVersion`	**Field Type** double **Description** Required. The version of the FlexCard or OmniScript.
`developerName`	**Field Type** string **Description** Required. The unique name of the OmniTrackingComponentDef in the API. This name can contain only underscores and alphanumeric characters and must be unique in your organization. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. Limit: 80 characters. Note: When creating large sets of data, always specify a unique `DeveloperName` for each record. If no `DeveloperName` is specified, performance may slow while Salesforce generates one for each record. Note: Only users with View DeveloperName OR View Setup and Configuration permission can view, group, sort, and filter this field.
`masterLabel`	**Field Type** string

Field Name	Description
	Description Required. The unique master label of the OmniTrackingComponentDef. This internal label doesn't get translated.
omniTrackingComponentDefKey	**Field Type** string **Description** A UUID generated internally by Salesforce to uniquely identify an OmniTrackingComponentDef record across all orgs.
omniTrackingGroup	**Field Type** string **Description** The ID of the related OmniTrackingGroup object.

Declarative Metadata Sample Definition

The following is an example of an OmniTrackingGroup component.

```
<?xml version="1.0" encoding="UTF-8"?>
<OmniTrackingGroup xmlns="http://soap.sforce.com/2006/04/metadata">
    <developerName>Purchase_Tracking</developerName>
    <groupType>Internal</groupType>
    <isActive>true</isActive>
    <masterLabel>Purchase_Tracking</masterLabel>
    <omniTrackingComponentDefs>
        <componentType>Omniscript</componentType>
        <componentVersion>2</componentVersion>
        <developerName>Purchase_Funnel</developerName>
        <masterLabel>Purchase_Funnel</masterLabel>
    </omniTrackingComponentDefs>
</OmniTrackingGroup>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>OmniTrackingGroup</name>
    </types>
    <version>60.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

OutboundNetworkConnection

Represents a private connection between a Salesforce org and a third-party data service. The connection is outbound because the callouts are going *out* of Salesforce. This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

OutboundNetworkConnection components have the suffix `.outboundNetworkConnection` and are stored in the `outboundNetworkConnections` folder.

Version

OutboundNetworkConnection components are available in API version 49.0 and later.

Fields

Field Name	Field Type	Description
`connectionType`	ExternalConnectionType (enumeration of type string)	Required. Specifies the cloud provider of the connection. • `AwsPrivateLink` • `DataCloudPrivateConnection` (Reserved for internal use)
`description`	string	A description of the connection. Maximum of 255 characters.
`isActive`	boolean	Required. Specifies whether the connection is active (`true`) or not (`false`).
`label`	string	Required. A user-friendly label for the connection.
`outboundNetworkConnProperties`	OutboundNetwokConnProperty on page 1559[]	Name-value pairs that describe the properties of an outbound network connection. Specify a name-value pair for each of the properties.
`status`	ExternalConnectionStatus (enumeration of type string)	Required. Connection status. The connection is initially Unprovisioned and moves through the other statuses automatically after an admin performs a Provision, Sync, or Teardown action. The valid values are: • `Unprovisioned` • `Allocation` • `PendingAcceptance` • `PendingActivation` • `RejectedRemotely` • `DeletedRemotely`

Field Name	Field Type	Description
		• `TeardownInProgress` • `Ready`

OutboundNetworkConnProperty

Represents a name-value pair that describes the properties of an outbound network connection.

Field Name	Field Type	Description
`propertyName`	OutboundConnPropertyName (enumeration of type string)	Required. The name of a property used to establish to an OutboundNetworkConnection. Valid values are: • `AwsVpcEndpointId`—The unique endpoint ID provided by Salesforce after an outbound AwsPrivateLink is created. The value is read-only when the `status` is `Ready`. • `AwsVpcEndpointServiceName`—The name of the customer's endpoint service running in an AWS VPC that's used for private connections with Salesforce. • `Region`—The region in which the VPC is hosted. Enumerated values `DataCloudPrivateNetwork*` are reserved for internal use.
`propertyValue`	`string`	Required. The value of OutboundConnPropertyName. For example, the `propertyValue` of `Region` can be `us-west-2`.

Declarative Metadata Sample Definition

The following sample definition has the suffix `.outboundNetworkConnection`.

```
<?xml version="1.0" encoding="UTF-8"?>
<OutboundNetworkConnection xmlns="http://soap.sforce.com/2006/04/metadata">
    <connectionType>AwsPrivateLink</connectionType>
    <description>Outbound Connection to make a callout to a Service deployed in AWS
VPC</description>
    <isActive>true</isActive>
    <label>MyOutboundConnection</label>
    <outboundNetworkConnProperties>
        <propertyName>Region</propertyName>
        <propertyValue>us-west-2</propertyValue>
    </outboundNetworkConnProperties>
    <outboundNetworkConnProperties>
        <propertyName>AwsVpcEndpointServiceName</propertyName>

<propertyValue>com.amazonaws.vpce.us-west-2.vpce-svc-00d7bd6285c123b4c</propertyValue>
    </outboundNetworkConnProperties>
    <status>Unprovisioned</status>
</OutboundNetworkConnection>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <fullName>sampleOutboundConnection</fullName>
    <types>
        <members>MyOutboundConnection</members>
        <name>OutboundNetworkConnection</name>
    </types>
    <version>49.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

Package

Specifies which metadata components to retrieve as part of a `retrieve()` call or defines a package of components.

Name	Type	Description
`apiAccessLevel`	APIAccessLevel (enumeration of type string)	Package components have access via dynamic Apex and the API to standard and custom objects in the organization where they're installed. Administrators who install packages can restrict this access after installation for improved security. The valid values are: • Unrestricted—Package components have the same API access to standard objects as the user who is logged in when the component sends a request to the API. • Restricted—The administrator can select which standard objects the components can access. Further, the components in restricted packages can only access custom objects in the current package if the user's permissions allow access to them. For more information, see "API and Dynamic Apex Access in Packages" in Salesforce Help.
`description`	string	A short description of the package.
`fullName`	string	The package name used as a unique identifier for API access. The `fullName` can contain only underscores and alphanumeric characters. It must be unique, begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. This field is inherited from the Metadata component.

Name	Type	Description
`namespacePrefix`	string	The namespace of the developer organization where the package was created.
`objectPermissions`	ProfileObjectPermissions[]	Indicates which objects are accessible to the package, and the kind of access available (create, read, update, delete).
`packageType`	string	Reserved for future use.
`postInstallClass`	string	The name of the Apex class that specifies the actions to execute after the package has been installed or upgraded. The Apex class must be a member of the package and must implement the Apex `InstallHandler` interface. In patch upgrades, you can't change the class name in this field but you can change the contents of the Apex class. The class name can be changed in major upgrades. This field is available in API version 24.0 and later.
`setupWeblink`	string	The weblink used to describe package installation.
`types`	PackageTypeMembers on page 1561[]	The type of component being retrieved.
`uninstallClass`	string	The name of the Apex class that specifies the actions to execute after the package has been uninstalled. The Apex class must be a member of the package and must implement the Apex `UninstallHandler` interface. In patch upgrades, you can't change the class name in this field but you can change the contents of the Apex class. The class name can be changed in major upgrades. This field is available in API version 25.0 and later.
`version`	string	Required. The version of the component type.

PackageTypeMembers

Use to specify the name and type of components to be retrieved in a package.

Name	Type	Description
`members`	string	One or more named components, or the wildcard character (`*`) to retrieve all metadata components of the type specified in the `<name>` element. To retrieve a standard object, specify it by name. For example, `<members>Account</members>` retrieves the standard Account object.

Name	Type	Description
`name`	string	The type of metadata component to be retrieved. For example, `<name>CustomObject</name>` retrieves one or more custom objects as specified in the `<members>` element.

Wildcard Support in the Manifest File

This metadata type doesn't support the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SEE ALSO:

Sample package.xml Manifest Files

ParticipantRole

Represents details, such as the name and associated default access level, for a role that a participant can have in the context of a parent record.

[other]: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

`ParticipantRole` components have the suffix `.participantRole` and are stored in the `participantRoles` folder.

Version

ParticipantRole components are available in API version 50.0 and later.

Fields

Field Name	Description
`defaultAccessLevel`	**Field Type** picklist **Description** Required. The default sharing access granted to the participant role. Valid values are:

Field Name	Description
	• `Edit`—Read/Write • `None` • `Read`—Read Only
`isActive`	**Field Type** boolean **Description** Indicates whether the participant role is activated.
`masterLabel`	**Field Type** string **Description** Required. The name for the participant role.
`parentObject`	**Field Type** string **Description** Required. The parent object for the participant role. Valid values are: • `Account` • `Budget` Available in API version 59.0 and later. • `IndividualApplication` Available in API version 59.0 and later. • `Interaction` Available in API version 52.0 and later. • `InteractionSummary` Available in API version 51.0 and later. • `FinancialDeal` Available in API version 52.0 and later. • `FundingAward` Available in API version 59.0 and later. • `FundingOpportunity` • `Opportunity` • `Team` Available in API version 58.0 and later. • Custom objects

Declarative Metadata Sample Definition

The following is an example of a ParticipantRole component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ParticipantRole xmlns="http://soap.sforce.com/2006/04/metadata">
    <defaultAccessLevel>Read</defaultAccessLevel>
    <isActive>true</isActive>
    <masterLabel>Advisor</masterLabel>
    <parentObject>Account</parentObject>
</ParticipantRole>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>ParticipantRole</name>
    </types>
    <version>50.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

PathAssistant

Represents Path records.This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Note the following when working with PathAssistant:

- Only one path can be created per record type for each object, including __Master__ record type.
- Rich text guidance information cannot be retrieved or deployed from or to translation workbench.
- The preference does not need to be on to retrieve or deploy PathAssistant.

File Suffix and Directory Location

PathAssistant components have the suffix `.pathAssistant` and are stored in the `pathAssistants` folder.

Version

PathAssistant components are available in API version 34.0 and later.

Fields

Field Name	Field Type	Description
`active`	boolean	Indicates whether the path is active (`true`) or not (`false`).
`entityName`	string	Required. The entity name. This is hard coded for Opportunity, Lead, and Quote. For a custom object, this field must be specified and should be the name of the custom object. This field is not updateable.
`fieldName`	string	Required. The field name. This is hard coded for StageName and Status. For a custom object, this field must be specified and should be the name of the picklist field that determines the steps in the path. This field is not updateable.
`masterLabel`	string	Required. The label of the path.
`pathAssistantSteps`	PathAssistantStep[] on page 1565	List of all the steps that have been configured with fields and guidance information. Note that a missing step in the .xml file means it has not been configured, not that it doesn't exist.
`recordTypeName`	string	Required. The name of the record type associated with the path. This field is not updateable.

PathAssistantStep

Represents the steps or stages in a Path.

Field Name	Field Type	Description
`fieldNames`	string	All the fields in `entityName` that will display in this step.
`info`	string	The guidance information displayed in this step.
`picklistValueName`	string	Required. The picklist value associated with the step.

Declarative Metadata Sample Definition

The following is an example of a PathAssistant component.

```
<?xml version="1.0" encoding="UTF-8"?>
<PathAssistant xmlns="http://soap.sforce.com/2006/04/metadata">
    <active>true</active>
    <entityName>Opportunity</entityName>
    <fieldName>StageName</fieldName>
    <masterLabel>Test Path</masterLabel>
    <pathAssistantSteps>
        <fieldNames>Amount</fieldNames>
        <fieldNames>CloseDate</fieldNames>
        <info>Some Text</info>
        <picklistValueName>Id. Decision Makers</picklistValueName>
    </pathAssistantSteps>
```

```
    <pathAssistantSteps>
        <fieldNames>Amount</fieldNames>
        <fieldNames>CloseDate</fieldNames>
        <info>Some Text</info>
        <picklistValueName>Proposal/Price Quote</picklistValueName>
    </pathAssistantSteps>
    <recordTypeName>Test_Record_Type</recordTypeName>
</PathAssistant>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
  <types>
        <members>Opportunity.Test_Busines_Process</members>
        <name>BusinessProcess</name>
    </types>
    <types>
        <members>Opportunity.StageName</members>
        <members>Lead.LeadSource</members>
        <members>Opportunity.Type</members>
        <name>CustomField</name>
    </types>
    <types>
        <members>Test_Path</members>
        <name>PathAssistant</name>
    </types>
    <types>
        <members>Opportunity.Test_Record_Type</members>
        <name>RecordType</name>
    </types>
    <types>
        <members>PathAssistant</members>
        <name>Settings</name>
    </types>
    <version>64.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

PaymentGatewayProvider

Represents the metadata associated with a payment gateway provider. This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

PaymentGatewayProvider components have the suffix `paymentGatewayProvider` and are stored in the `paymentGatewayProviders` folder.

Version

PaymentGatewayProvider components are available in API version 48.0 and later.

Special Access Rules

To access PaymentGatewayProvider, you must have a Salesforce Order Management license with the PaymentPlatform org permission activated.

Fields

Field Name	Field Type	Description
`apexAdapter`	string	The Apex adapter class name for your payment gateway. This field is unique within your organization.
`comments`	string	Users can add comments to provide additional details about a record. Maximum of 1000 characters.
`idempotencySupported`	idempotencySupportStatus (enumeration of type String)	Required. Defines whether the payment gateway ignores duplicate payment gateway calls (`Yes`) or whether it processes duplicate gateway calls (`No`). • `Yes` • `No`
`masterLabel`	string	Required. The label of this payment gateway provider record.

Declarative Metadata Sample Definition

The following is an example of a PaymentGatewayProvider component.

```
<PaymentGatewayProvider xmlns="http://soap.sforce.com/2006/04/metadata">
    <apexAdapter>SalesforceAdapter</apexAdapter>
    <idempotencySupported>Yes</idempotencySupported>
    <masterLabel>SalesforceAdapter</masterLabel>
    <comments>Comments</comments>
</PaymentGatewayProvider>
```

The following is an example `package.xml` that references the previous definition.

```
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>PaymentGatewayProvider</name>
```

```
    </types>
    <version>48.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

PermissionSet

Represents a set of permissions that's used to grant more access to one or more users without changing their profile or reassigning profiles. You can use permission sets to grant access but not to deny access.

This type extends the Metadata metadata type and inherits its `fullName` field.

In API version 40.0 and later, when you retrieve permission set metadata, all content exposed in Metadata API for the permission sets is included. The metadata includes Apex associated with the permission set, CRUD, and so on. Likewise, when you deploy a permission set, you must include all of its metadata to avoid accidentally overwriting the permission set's contents.

In API version 39.0 and earlier, retrieving or deploying permission set metadata returns only app and system permissions assigned to the permission set. Junction metadata (such as Apex, CRUD) are included only if the metadata for the related component is also included in the package definition.

In API version 29.0 and later, you can retrieve and deploy access settings for these managed components in profiles and permission sets:

For more information, see the Managed Component Access section of Sample package.xml Manifest Files in the *Metadata API Developer Guide*.

Declarative Metadata File Suffix and Directory Location

Permission sets are stored in the `permissionsets` directory. The file name matches the permission set API name and the extension is `.permissionset`. For example, a permission set with the name *User_Management_Perms* is stored in `permissionsets/User_Management_Perms.permissionset`.

Version

Permission sets are available in API version 22.0 and later.

Special Access Rules

As of Summer '20 and later, only users who have one of these permissions can access this type:

- View Setup and Configuration
- Manage Session Permission Set Activations
- Assign Permission Sets
- Manage Profiles and Permission Sets

To view the following settings, assignments, and permissions for standard and custom objects in a specified permission set, the View Setup and Configuration permission is required.

- Client settings
- Field permissions
- Layout assignments
- Object permissions
- Permission dependencies
- Permission set tab settings
- Permission set group components
- Record types

Fields

Field	Field Type	Description
`agentAccesses`	PermissionSetAgentAccess[]	Indicates which agents are visible to users assigned to this permission set. Available in API version 63.0 and later.
`applicationVisibilities`	PermissionSetApplicationVisibility[]	Indicates which apps are visible to users assigned to this permission set. Available in API version 29.0 and later. In API version 29.0, this field supports custom apps only. In API version 30.0 and later, this field supports both standard and custom apps.
`classAccesses`	PermissionSetApexClassAccess[]	Indicates which top-level Apex classes have methods that users assigned to this permission set can execute. Available in API version 23.0 and later.
`customMetadataTypeAccesses`	PermissionSetCustomMetadataTypeAccess[]	Indicates the custom metadata types that are read-accessible to a user assigned to this permission set. Available in API version 47.0 and later.
`customPermissions`	PermissionSetCustomPermissions[]	Indicates which custom permissions are available to users assigned to this permission set. Available in API version 31.0 and later.
`customSettingAccesses`	PermissionSetCustomSettingAccesses[]	Indicates the custom settings that are read-accessible to a user assigned to this permission set. Available in API version 47.0 and later.
`description`	string	The permission set description. Limit: 255 characters.
`emailRoutingAddressAccesses`	PermissionSetEmailRoutingAddressAccess[]	Indicates the Email Routing Address permissions that are available to users assigned to a permission set. Available in API version 62.0 and later.
`externalCredentialPrincipalAccesses`	PermissionSetExternalCredentialPrincipalAccess[]	Indicates which external credential principals are available to users assigned to this permission set. Available in API version 59.0 and later.
`externalDataSourceAccesses`	PermissionSetExternal DataSourceAccess[]	Indicates which data sources with identity type of `Per User` are available to users assigned to this permission set. Available in API version 27.0 and later.

Field	Field Type	Description
`fieldPermissions`	PermissionSetFieldPermissions[]	Indicates which fields are accessible to a user assigned to this permission set, and the kind of access available (readable or editable). Available in API version 23.0 and later.
`flowAccesses`	PermissionSetFlowAccess[]	Indicates which flows can be accessed by a user assigned to this permission set. Available in API version 47.0 and later.
`hasActivationRequired`	boolean	Indicates whether the permission set requires an associated active session (`true`) or not (`false`). Available in API version 37.0 and later.
`label`	string	Required. The permission set label. Limit: 80 characters.
`license`	string	Either the related permission set license or the user license associated with this permission set. Available in API version 38.0 and later. Use this field instead of `userLicense`, which is deprecated and only available up to API Version 37.0.
`objectPermissions`	PermissionSetObjectPermissions[]	Indicates the objects that are accessible to a user assigned to this permission set, and the kind of access available (create, read, edit, delete, and so on). Available in API version 23.0 and later.
`pageAccesses`	PermissionSetApexPageAccess[]	Indicates which Visualforce pages that users assigned to this permission set can execute. Available in API version 23.0 and later.
`recordTypeVisibilities`	PermissionSetRecordTypeVisibility[]	Indicates which record types are visible to users assigned to this permission set. Available in API version 29.0 and later. This field is never retrieved or deployed for inactive record types.
`ServicePresenceStatusAccesses`	PermissionSetServicePresenceStatusAccess[] on page 1576	Indicates which Service presence statuses that the user assigned to this profile can execute. Available in API version 64.0 and later.
`tabSettings`	PermissionSetTabVisibility[]	Indicates the tab visibility settings for this permission set. Available in API version 26.0 and later.
`userLicense`	string	Deprecated. The user license for the permission set. A user license determines the baseline of features that the user can access. Every user must have exactly one user license. Available up to API version 37.0. In API version 38.0 and later, use `license`.
`userPermissions`	PermissionSetUserPermissions[]	Specifies an app or system permission (such as "API Enabled") and whether it's enabled for this permission set. In API version 28.0 and earlier, this field retrieves all user permissions, enabled or disabled. In API version 29.0

Field	Field Type	Description
		and later, this field retrieves only enabled user permissions. In API Version 40.0 and later, if a permission isn't specified for a deployment, it's disabled.

PermissionSetAgentAccess

PermissionSetAgentAccess represents the agent access configuration for users assigned through a permission set.

Field Name	Field Type	Description
`agentName`	string	Required. The name of the employee agent.
`enabled`	boolean	Required. Indicates whether users assigned to this permission set can use the Agentforce Employee Agent (`true`) or not (`false`).

PermissionSetApplicationVisibility

PermissionSetApplicationVisibility on page 1571 determines whether an app is visible to a user assigned to this permission set.

Field Name	Field Type	Description
`application`	string	Required. The app name.
`visible`	boolean	Required. Indicates whether this app is visible to users assigned to this permission set (`true`) or not (`false`).

PermissionSetApexClassAccess

PermissionSetApexClassAccess on page 1571 represents the Apex class access for users assigned to a permission set.

Field	Field Type	Description
`apexClass`	string	Required. The Apex class name.
`enabled`	boolean	Required. Indicates whether users assigned to this permission set can execute methods in the top-level class (`true`) or not (`false`).

PermissionSetCustomMetadataTypeAccess

PermissionSetCustomMetadataTypeAccess on page 1571 represents the custom metadata type access for users assigned to a permission set. Available in API version 47.0 and later.

Field	Field Type	Description
`enabled`	boolean	Required. Indicates whether the records for this custom metadata type are readable (`true`) or not (`false`).

Field	Field Type	Description
`name`	string	Required. The custom metadata type name.

PermissionSetCustomPermissions

PermissionSetCustomPermissions represents the custom permissions access for users assigned to a permission set. Only enabled custom permissions are retrieved.

Field Name	Field Type	Description
`enabled`	boolean	Required. Indicates whether the custom permission is enabled (`true`) or not (`false`).
`name`	string	Required. The custom permission name.

PermissionSetCustomSettingAccesses

PermissionSetCustomSettingAccesses represents the custom setting access for users assigned to a permission set. Available in API version 47.0 and later.

Field	Field Type	Description
`enabled`	boolean	Required. Indicates whether the records for this custom setting are readable (`true`) or not (`false`).
`name`	string	Required. The custom setting name.

PermissionSetEmailRoutingAddressAccess

PermissionSetEmailRoutingAddressAccess represents the Email Routing Address access for users assigned to a permission set. Only enabled email routing addresses are retrieved.

Field	Field Type	Description
`enabled`	boolean	Required. Indicates whether the custom permission is enabled (`true`) or not (`false`).
`name`	string	Required. Represents an organization's Email-to-Case routing address.

PermissionSetExternalCredentialPrincipalAccess

PermissionSetExternalCredentialPrincipalAccess on page 1572 represents the access to the external credential's principals. Users assigned to the permission set can make callouts using a named credential that references the external credential. Available in API version 59.0 and later.

Field	Field Type	Description
`enabled`	boolean	Required. Indicates whether external credential principal access is enabled on the permission set (`true`) or not (`false`).
`externalCredentialPrincipal`	string	Required. The name of the external credential and principal, separated by a dash. For example, `myExternalCredential-myPrincipal`. If the external credential and principal are part of a package, include the package's namespace prefix with the principal's name using this format: `namespacePrefix__myExternalCredential-myPrincipal`. Use two underscores (__) between the namespace prefix and the external credential principal's name.

PermissionSetExternalDataSourceAccess

PermissionSetExternalDataSourceAccess on page 1573 represents the data source access for users with identity type of `Per User`. Available in API version 27.0 and later.

Field	Field Type	Description
`enabled`	boolean	Required. Indicates whether the data source is enabled (`true`) or not (`false`).
`externalDataSource`	string	The name of the external data source.

PermissionSetFieldPermissions

PermissionSetFieldPermissions on page 1573 represents the field permissions for users assigned to a permission set. In API version 30.0 and later, permissions for required fields can't be retrieved or deployed. In API version 54.0 and later, only field permissions enabled in the permission set are returned in queries.

As of API version 38.0, you can change field permissions to make a field editable using the Metadata API for fields that you can't change through the user interface. For example, you can deploy `Asset.ProductCode` as an editable field even though you can't through the user interface.

Note: If the View All Fields object permission is enabled for an object in the permission set, the individual fields aren't returned under `fieldPermissions`. However, if you later disable the View All Fields object permission, the fields are returned under `fieldPermissions` and you can remove access to the fields manually.

Field	Field Type	Description
`editable`	boolean	Required. Indicates whether the field can be edited by the users assigned to this permission set (`true`) or not (`false`).

Field	Field Type	Description
field	string	Required. The API name of the field (such as Warehouse__c.Description__c). When referencing shared Activity fields, specify Event or Task. For example, Event.Meeting__c.
readable	boolean	Indicates whether the field can be read by the users assigned to this permission set (true) or not (false).

PermissionSetFlowAccess

PermissionSetFlowAccess on page 1574 represents which flows a permission set grants access to. Available in API version 47.0 and later.

Field	Field Type	Description
enabled	boolean	Required. Indicates whether users assigned this permission set can access the flow (true) or not (false) The default value is false.
flow	string	Required. The name of the flow to which access is granted.

PermissionSetObjectPermissions

PermissionSetObjectPermissions represents the object permissions for a permission set. Use one of these elements for each permission.

Field	Field Type	Description
allowCreate	boolean	Required. Indicates whether the object referenced by the object field can be created by the users assigned to this permission set (true) or not (false).
allowDelete	boolean	Required. Indicates whether the object referenced by the object field can be deleted by the users assigned to this permission set (true) or not (false).
allowEdit	boolean	Required. Indicates whether the object referenced by the object field can be edited by the users assigned to this permission set (true) or not (false).
allowRead	boolean	Required. Indicates whether the object referenced by the object field can be viewed by the users assigned to this permission set (true) or not (false).
modifyAllRecords	boolean	Required. Indicates whether all records for the object referenced by the object field can be viewed, edited, or deleted by the users assigned to this permission set (true) or not (false), regardless of the sharing settings for the object. Includes private records (records with no parent object). Similar to the Modify All Data user permission, but limited to the individual object level.

Field	Field Type	Description
`object`	string	Required. The API name of the object (such as `Warehouse__c`).
`viewAllFields`	string	Indicates whether all fields and field data for the object referenced by the `object` field can be viewed by the users assigned to this permission set (`true`) or not (`false`). Available in API version 63.0 and later.
`viewAllRecords`	boolean	Required. Indicates whether all records for the object referenced by the `object` field can be viewed by the users assigned to this permission set (`true`) or not (`false`), regardless of the sharing settings for the object. This setting includes private records (records with no parent object). The `viewAllRecords` field is similar to the View All Data user permission but limited to the individual object level.

PermissionSetApexPageAccess

PermissionSetApexPageAccess on page 1575 represents the Visualforce page access for users assigned to a permission set.

Field	Field Type	Description
`apexPage`	string	Required. The Visualforce page name.
`enabled`	boolean	Required. Indicates whether users assigned to this permission set can execute the Visualforce page (`true`) or not (`false`).

PermissionSetRecordTypeVisibility

PermissionSetRecordTypeVisibility on page 1575 represents the visibility of record types for this permission set.

Field	Field Type	Description
`recordType`	string	Required. The record type name, for example `Account.MyRecordType`.
`visible`	boolean	Required. Indicates whether the record type is visible to users assigned to this permission set (`true`) or not (`false`).

PermissionSetTabSetting

PermissionSetTabSetting on page 1575 represents the tab settings for a permission set.

Field	Field Type	Description
`tab`	string	Required. The tab name.

Field	Field Type	Description
visibility	PermissionSetTabVisibility (enumeration of type string)	Required. Indicates the visibility settings for the tab. Valid values are: • Available—The tab is available on the All Tabs page. Individual users can customize their display to make the tab visible in any app. • None—The tab isn't available on the All Tabs page or visible in any apps. • Visible—The tab is available on the All Tabs page and appears in the visible tabs for its associated app. Individual users can customize their display to hide the tab or make it visible in other apps.

PermissionSetUserPermission

In API version 28.0 and earlier, PermissionSetUserPermission represents an app or system permission for a permission set. In API version 29.0 and later, this field retrieves only enabled user permissions. Use one of these elements for each permission.

Field	Field Type	Description
enabled	boolean	Required. Indicates whether the permission is enabled (true) or disabled (false).
name	string	Required. The name of the permission.

PermissionSetServicePresenceStatusAccess

Represents the presence statuses that reps assigned to this profile have access. Available in API version 64.0 and later.

Field	Field Type	Description
servicePresenceStatus	string	Required. The name of Service Presence Status.
enabled	boolean	Required. Indicates whether the rep assigned to this profile has access to the presence status (true) or not (false).

Declarative Metadata Sample Definition

The following is an example of a PermissionSet component.

```
<?xml version="1.0" encoding="UTF-8"?>
<PermissionSet xmlns="http://soap.sforce.com/2006/04/metadata">
    <description>Grants all rights needed for an HR administrator to manage
employees.</description>
    <label>HR Administration</label>
    <userLicense>Salesforce</userLicense>
    <applicationVisibilities>
```

```
        <application>JobApps__Recruiting</application>
        <visible>true</visible>
    </applicationVisibilities>
    <userPermissions>
        <enabled>true</enabled>
        <name>ApiEnabled</name>
    </userPermissions>
    <objectPermissions>
        <allowCreate>true</allowCreate>
        <allowDelete>true</allowDelete>
        <allowEdit>true</allowEdit>
        <allowRead>true</allowRead>
        <viewAllRecords>true</viewAllRecords>
        <modifyAllRecords>true</modifyAllRecords>
        <viewAllFields>true</viewAllFields>
        <object>Job_Request__c</object>
    </objectPermissions>
    <fieldPermissions>
        <editable>true</editable>
        <field>Job_Request__c.Salary__c</field>
        <readable>true</readable>
    </fieldPermissions>
    <pageAccesses>
        <apexPage>Job_Request_Web_Form</apexPage>
        <enabled>true</enabled>
    </pageAccesses>
    <classAccesses>
      <apexClass>Send_Email_Confirmation</apexClass>
      <enabled>true</enabled>
    </classAccesses>
    <tabSettings>
        <tab>Job_Request__c</tab>
        <visibility>Available</visibility>
    </tabSettings>
    <recordTypeVisibilities>
        <recordType>Recruiting.DevManager</recordType>
        <visible>true</visible>
    </recordTypeVisibilities>
</PermissionSet>
```

The following is an example package.xml manifest used to retrieve the PermissionSet metadata for an organization. When you retrieve permission sets, also retrieve the related components with assigned permissions. For example, to retrieve `objectPermissions` and `fieldPermissions` for a custom object, you must also retrieve the CustomObject component.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>Job_Request__c</members>
        <name>CustomTab</name>
    </types>
    <types>
        <members>Job_Request__c</members>
        <name>CustomObject</name>
    </types>
    <types>
```

```
        <members>JobApps__Recruiting</members>
        <name>CustomApplication</name>
    </types>
    <types>
        <members>Recruiting.DevManager</members>
        <name>RecordType</name>
    </types>
    <types>
        <members>*</members>
        <name>PermissionSet</name>
    </types>
    <version>64.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

PermissionSetGroup

Represents a group of permission *sets* and the permissions within them. Use permission set groups to organize permissions based on job functions or tasks. Then, you can package the groups as needed.

This type extends the Metadata metadata type and inherits its `fullName` field.

Declarative Metadata File Suffix and Directory Location

Permission set groups are stored in the `permissionsetgroups` directory. The file name matches the permission set API name and the extension is `.permissionsetgroup`. For example, a permission set group with the name `Finance_Mgmt_PermSetGroup` is stored in `permissionsetgroups/Finance_Mgmt_PermSetGroup.permissionsetgroup`.

Version

Permission set groups are available in API version 45.0 and later.

Special Access Rules

As of Summer '20 and later, to view this type, users must have one of these permissions:

- View Setup and Configuration
- Manage Session Permission Set Activations
- Assign Permission Sets

To edit this type, users must have the Manage Profiles and Permission Sets permission.

Fields

Field	Field Type	Description
`description`	string	The permission set group description provided by the permission set group creator.
`hasActivationRequired`	boolean	Indicates whether the permission set group requires an associated active session (`true`) or not (`false`). The default value is `false`. This field is available in API version 53.0 and later.
`label`	string	Required. The permission set group label.
`mutingPermissionSets`	string	A permission set containing permissions to disable in the permission set group. This field is available in API version 46.0 and later.
`permissionSets`	string	A permission set or permission sets included in the permission set group.
`status`	string	Indicates permission set group recalculation status. Valid values are: • `Updated`—The group is current. • `Outdated`—The group requires recalculation. • `Updating`—The group is in recalculation mode. • `Failed`—The group recalculation failed.

Declarative Metadata Sample Definition

When adding a permission set group, you can do something like this. Individual permissions are included in the permission set referenced, not in the permission set group.

```
<?xml version="1.0" encoding="UTF-8"?>
<PermissionSetGroup xmlns="http://soap.sforce.com/2006/04/metadata">
     <fullName>Finance_Mgmt_PermSetGroup</fullName>
     <description>Finance_Mgmt_PermSetGroup desc</description>
     <label>Finance_Mgmt_PermSetGroup</label>
     <permissionSets>Billing_PS</permissionSets>
</PermissionSetGroup>
```

The permission set `Billing_PS` contains the individual permissions included in `Finance_Mgmt_PermSetGroup`.

```
<?xml version="1.0" encoding="UTF-8"?>
<PermissionSet xmlns="http://soap.sforce.com/2006/04/metadata">
     <fullName>Billing_PS</fullName>
     <description>Billing_PS</description>
     <label>Billing_PS</label>
     <hasActivationRequired>false</hasActivationRequired>
     <license>Salesforce</license>
     <userPermissions>
```

```
        <enabled>true</enabled>
        <name>ViewSetup</name>
    </userPermissions>
    <userPermissions>
        <enabled>true</enabled>
        <name>ViewRoles</name>
    </userPermissions>
    <userPermissions>
        <enabled>true</enabled>
        <name>EditBillingInfo</name>
   </userPermissions>
</PermissionSet>
```

This example `package.xml` manifest retrieves the PermissionSetGroup metadata for an org. When you retrieve permission set groups, also retrieve the related components. For example, to retrieve PermissionSetGroup, you must also retrieve PermissionSet.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>Finance_Mgmt_PermSetGroup</members>
        <name>PermissionSetGroup</name>
    </types>
    <types>
        <members>Billing_PS</members>
        <name>PermissionSet</name>
    </types>
    <version>45.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

PermissionSetLicenseDefinition (Developer Preview)

Represents the definition of a custom permission set license, which entitles specified features in a package.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

`PermissionSetLicenseDefinition` components have the suffix `.permissionSetLicenseDefinition` and are stored in the `permissionSetLicenseDefinitions` folder.

Version

PermissionSetLicenseDefinition components are available in API version 54.0 and later.

Special Access Rules

To access PermissionSetLicenseDefinition, you must have the Partner Licensing Platform developer preview enabled. To participate in this developer preview, submit a participation request via the Partner Licensing Platform Developer Preview Partner Community group.

Note: The Partner Licensing Platform is available as a developer preview. The Partner Licensing Platform isn't generally available unless or until Salesforce announces its general availability in documentation or in press releases or public statements. All commands, parameters, and other features are subject to change or deprecation at any time, with or without notice. Don't implement functionality developed with these commands or tools in your production package.

Fields

Field Name	Field Type	Description
`customPermissions`	PermissionSetLicenseDefinitionCustomPermission	An array of licensed custom permissions included in the permission set license definition.
`isSupplementLicense`	boolean	Indicates whether the custom permission set license is a supplement license (`true`) or a foundation license (`false`). The default value is `false`. This field is available in API version 55.0 and later.
`label`	string	Required. The name of the permission set license definition.
`licenseExpirationPolicy`	LicenseExpirationPolicy (enumeration of type string)	The license expiration policy of the custom permission set license. Valid values are: • `BlockNamespaceAccess`—Package access is blocked for existing users when all custom permission set licenses expire. This is the default value. • `AllowNamespaceAccess`—Package access isn't blocked for existing users when all custom permission set licenses expire. This field is available in API version 55.0 and later.
`userLicenseRestrictions`	string	The user license categories that can be assigned the custom permission set license. If no user license categories are specified, all users can be assigned the license. Possible values include: • `${communities}` • `${communitiesLogin}` • `${customerCommunities}` • `${customerCommunitiesLogin}` • `${internal}` • `${partnerCommunity}` • `${partnerCommunityLogin}` • `${platform}`

Field Name	Field Type	Description
		For more information, see User License Restriction Categories (Developer Preview). This field is available in API version 55.0 and later.

PermissionSetLicenseDefinitionCustomPermission

Represents a licensed custom permission included in the permission set license definition.

Field Name	Field Type	Description
`name`	string	Label of the licensed custom permission. This field must be a reference to a CustomPermission that has the `isLicensed` field set to `true`.

Declarative Metadata Sample Definition

The following is an example of a PermissionSetLicenseDefinition component.

```
<?xml version="1.0" encoding="UTF-8"?>
<PermissionSetLicenseDefinition xmlns="http://soap.sforce.com/2006/04/metadata">
    <customPermissions>>
        <name>AccessReportsPerm</name>
    </customPermissions>
    <isSupplementLicense>false</isSupplementLicense>
    <licenseExpirationPolicy>BlockNamespaceAccess</licenseExpirationPolicy>
    <label>ExampleFeatureLicenseDefinition</label>
    <userLicenseRestrictions>${internal}</userLicenseRestrictions>
</PermissionSetLicenseDefinition>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>PermissionSetLicenseDefinition</name>
    </types>
    <version>54.0</version>
</Package>
```

Usage

For more information, see the Partner Licensing Platform Developer Guide (Developer Preview).

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

PersonAccountOwnerPowerUser

Represents a user who can own more than 50,000 customer or partner portal accounts. Person account owner power users can own a large number of either customer or partner users. They can't change their role, look up to a parent role, or reparent their role. Person account owner power user objects can't be created if deferred sharing is turned on for your org. This object is available in API version 57.0 and later.

Version

PersonAccountOwnerPowerUser components are available in API version 57.0 and later.

Fields

Field Name	Description
`developerName`	**Field Type** string **Description** Required. The unique name of the object in the API.
`masterLabel`	**Field Type** string **Description** Required. The label entered when the person account owner power user is created.
`portalType`	**Field Type** string **Description** Required. The type of portal user account that the person account owner power user can own. Possible values are: • `CustomerPortal`—Customer Portal • `Partner`—Partner Portal
`user`	**Field Type** string **Description** Required. The unique ID associated with the person account owner power user.

PipelineInspMetricConfig

Represents the settings of Pipeline Inspection forecast category metrics.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

PipelineInspMetricConfig components have the suffix `.pipelineInspMetricConfig` and are stored in the `pipelineInspMetricConfigs` folder.

Version

PipelineInspMetricConfig components are available in API version 57.0 and later.

Special Access Rules

Only users with the Customize Application or Modify All Data permission can access this type.

Fields

Field Name	Description
`isCumulative`	**Field Type** boolean **Description** Required. Read only. Indicates whether the metric is cumulative (`true`) or not (`false`). The default value is `true`.
`isProtected`	**Field Type** boolean **Description** Indicates whether the component is protected (`true`) or not (`false`). The default value is `false`.
`masterLabel`	**Field Type** string **Description** Required. Customized label of the Pipeline Inspection metric. Limit: 50 characters.
`metric`	**Field Type** PipelineInspectionMetric (enumeration of type string)

Field Name	Description
	Description Required. The Pipeline Inspection metric. Possible values are: • `BestCase` (available in API version 58.0 and later) • `ClosedLost` (available in API version 58.0 and later) • `ClosedWon` (available in API version 58.0 and later) • `Commit` (available in API version 58.0 and later) • `MostLikely` (available in API version 58.0 and later) • `OpenPipeline` (available in API version 58.0 and later) • `TotalPipeline` (available in API version 58.0 and later)

Declarative Metadata Sample Definition

The following is an example of a PipelineInspMetricConfig component.

```
<?xml version="1.0" encoding="UTF-8"?>
<PipelineInspMetricConfig xmlns="http://soap.sforce.com/2006/04/metadata">
    <isCumulative>true</isCumulative>
    <isProtected>false</isProtected>
    <masterLabel>Lost the opportunity</masterLabel>
    <metric>ClosedLost</metric>

</PipelineInspMetricConfig>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>PipelineInspMetricConfig</name>
    </types>
    <version>57.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

PlatformCachePartition

Represents a partition in the Platform Cache. This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

PlatformCachePartition components have the suffix `.cachePartition` and are stored in the `cachePartitions` folder.

Version

PlatformCachePartition components are available in API version 35.0 and later.

Special Access Rules

The "Author Apex" permission is required to deploy and retrieve PlatformCachePartition components.

Fields

Field Name	Field Type	Description
`description`	string	Describes the cache partition.
`isDefaultPartition`	boolean	Required. Indicates whether this cache partition is the default partition in your organization (`true`) or not (`false`).
`masterLabel`	string	Required. The label of the cache partition that appears in the Salesforce user interface.
`platformCachePartitionTypes`	PlatformCachePartitionType[]	An array of cache types that the partition can store.

PlatformCachePartitionType

Contains information about a partition type, including its minimum and allocated capacity.

Field Name	Field Type	Description
`allocatedCapacity`	int	Required. The total storage capacity, in megabytes (MB), that is allocated for the cache type, including free, purchased, and trial cache. Purchased capacity includes organization-wide cache, which can be used in any partition, and namespace-specific cache, which can be used only in partitions associated with a namespace.
`allocatedPartnerCapacity`	int	Required. Free capacity, in megabytes (MB). allocated to Developer Edition orgs for the cache type. Use this capacity with security-reviewed managed packages. Available in API version 51.0 and later.
`allocatedPurchasedCapacity`	int	Required. The amount of namespace-specific purchased storage capacity, in MB, that is allocated for the cache type.
`allocatedTrialCapacity`	int	Required. The amount of trial cache space, in MB, that is allocated for the cache type.

Field Name	Field Type	Description
cacheType	PlatformCacheType (enumeration of type string)	The type of cache. Valid values are: • Session—Session cache • Organization—Org cache

Declarative Metadata Sample Definition

The following is an example of a PlatformCachePartition component.

```
<?xml version="1.0" encoding="UTF-8"?>
<PlatformCachePartition xmlns="http://soap.sforce.com/2006/04/metadata">
    <description>Custom partition and marked as default.</description>
    <isDefaultPartition>true</isDefaultPartition>
    <masterLabel>myPartition</masterLabel>
    <platformCachePartitionTypes>
        <allocatedCapacity>10</allocatedCapacity>
        <allocatedPurchasedCapacity>5</allocatedPurchasedCapacity>
        <cacheType>Session</cacheType>
    </platformCachePartitionTypes>
    <platformCachePartitionTypes>
        <allocatedCapacity>5</allocatedCapacity>
        <allocatedPurchasedCapacity>5</allocatedPurchasedCapacity>
        <cacheType>Organization</cacheType>
    </platformCachePartitionTypes>
</PlatformCachePartition>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>myPartition</members>
        <name>PlatformCachePartition</name>
    </types>
    <version>64.0</version>
</Package>
```

If a namespace is defined in your organization, add the namespace prefix to your partition name. For example:

```
<members>Namespace.myPartition</members>
```

To retrieve all cache partitions from your organization, use the wildcard character (*) as follows.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>PlatformCachePartition</name>
    </types>
    <version>64.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

PlatformEventChannel

Represents a channel that you can subscribe to in order to receive a stream of events. In API version 46.0 and earlier, it is the default standard channel for change data capture events. In API version 47.0 and later, it is a custom channel for change data capture events.

The default standard channel corresponds to the entity selection in the Change Data Capture page in Setup. A custom channel is a channel that you define using this metadata type. Starting in API version 47.0, the channel doesn't contain the selected entities, which are represented each by PlatformEventChannelMember. This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

PlatformEventChannel components have the suffix `.platformEventChannel` and are stored in the `platformEventChannels` folder.

Version

PlatformEventChannel components are available in API version 45.0 and later.

Special Access Rules

You must have the Customize Application permission to deploy and retrieve this type.

Fields

Field Name	Field Type	Description
`channelMembers`	PlatformEventChannelSelectedEntity[]	Removed. A list of event names of entities, including standard and custom objects, selected for Change Data Capture notifications. Note: This field is removed in API version 47.0 and later and is available only in API versions 45.0 and 46.0. In API version 47.0 and later, the channel members are each defined in a PlatformEventChannelMember component.
`channelType`	PlatformEventChannelType (enumeration of type string)	Required. The channel type. Valid values are: • `data`—Change Data Capture channel corresponding to the selected entities. • `event`—A channel that contains platform events.
`eventType`	PlatformEventChannelEventType	The type of events that the channel can hold. A channel can hold only one type of events. Use this field to optionally specify a specific type of

Field Name	Field Type	Description
	(enumeration of type string)	events for a channel in combination with the `channelType` field. Valid values are: • `custom`—The channel contains custom platform events. This value is valid with the `channelType` of `event`. • `data`—The channel contains change data capture events. This value is valid with the `channelType` of `data`. • `monitoring`—The channel contains Real-Time Event Monitoring events. This value is valid with the `channelType` of `event`. • `standard`—Reserved for internal use. Available in API version 61.0 and later.
`label`	string	Required. The channel label.

PlatformEventChannelSelectedEntity

Note: This field type is removed in API version 47.0 and later and is available only in API versions 45.0 and 46.0.

Field Name	Field Type	Description
`selectedEntity`	string	Required. The event name of an entity selected for Change Data Capture notifications. For example, for the Account standard object, the name is `AccountChangeEvent`, or for a custom object MyObject__c, the name is `MyObject__ChangeEvent`.

Usage

The createMetadata() and deleteMetadata() calls aren't supported with the PlatformEventChannel metadata type.

In API version 47.0 and later, you can't deploy or retrieve the ChangeEvents standard channel.

You can't delete the ChangeEvents standard channel with `destructiveChanges.xml`, but you can delete channel members using the PlatformEventChannelMember type with `destructiveChanges.xml`.

You can delete a custom channel with `destructiveChanges.xml`. If you delete a custom channel, all its member PlatformEventChannelMember components are also deleted.

You can update only the `fullName` field and the `label` field of a PlatformEventChannel component.

Declarative Metadata Sample Definition for a Custom Channel

The PlatformEventChannel component contains the label of the custom channel and the channel type.

```
<?xml version="1.0" encoding="UTF-8"?>
<PlatformEventChannel xmlns="http://soap.sforce.com/2006/04/metadata">
    <channelType>data</channelType>
    <label>Custom Channel for Sales Events</label>
</PlatformEventChannel>
```

This `package.xml` references the previous definition. The custom channel name is `SalesEvents__chn`.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>SalesEvents__chn</members>
        <name>PlatformEventChannel</name>
    </types>
    <version>64.0</version>
</Package>
```

Wildcard Support in the Manifest File

To deploy or retrieve all custom channels, specify the wildcard character `*` (asterisk) in the `<members>` field.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>PlatformEventChannel</name>
    </types>
    <version>64.0</version>
</Package>
```

Upgrading to Version 47.0 or Later From an Earlier Version

The `channelMembers` field of the PlatformEventChannel type is removed in API version 47.0 and later. As a result, PlatformEventChannel components created in prior versions can't be deployed using a later API version but you can deploy them in the same API version they were created with.

To deploy a custom channel component using API version 47.0 and later, upgrade the PlatformEventChannel definition by removing the `<channelMembers>` fields. For the ChangeEvents standard channel, it can't be deployed or retrieved, so delete the PlatformEventChannel definition file.

For example, if you had custom channel called SalesEvents__chn, this could be your custom channel definition in API version 46.0.

```
<?xml version="1.0" encoding="UTF-8"?>
<PlatformEventChannel xmlns="http://soap.sforce.com/2006/04/metadata">
    <channelMembers>
        <selectedEntity>AccountChangeEvent</selectedEntity>
    </channelMembers>
    <channelMembers>
        <selectedEntity>ContactChangeEvent</selectedEntity>
    </channelMembers>
    <channelType>data</channelType>
    <label>Sales Events</label>
</PlatformEventChannel>
```

To upgrade to version 47.0 or later, you would replace the custom channel definition with this definition, which doesn't contain any channel members.

```
<?xml version="1.0" encoding="UTF-8"?>
<PlatformEventChannel xmlns="http://soap.sforce.com/2006/04/metadata">
    <channelType>data</channelType>
```

```
    <label>SalesEvents__chn</label>
</PlatformEventChannel>
```

For each channel member that is part of either a custom or the standard ChangeEvents channel, add a PlatformEventChannelMember metadata component. Also, in the `package.xml` file, reference both the PlatformEventChannel and PlatformEventChannelMember components.

For example, this PlatformEventChannelMember component is for the AccountChangeEvent member.

```
<?xml version="1.0" encoding="UTF-8"?>
<PlatformEventChannelMember xmlns="http://soap.sforce.com/2006/04/metadata">
    <eventChannel>SalesEvents__chn</eventChannel>
    <selectedEntity>AccountChangeEvent</selectedEntity>
</PlatformEventChannelMember>
```

For more information, see PlatformEventChannelMember.

For an example of a custom channel that holds custom platform events and Real-Time Event Monitoring events, see Group Platform Events into One Stream with a Custom Channel in the *Platform Events Developer Guide*.

SEE ALSO:

Change Data Capture Developer Guide: Subscription Channels

Change Data Capture Developer Guide: Compose Streams of Change Data Capture Notifications with Custom Channels

PlatformEventChannelMember

PlatformEventChannelMember

Represents an entity selected for Change Data Capture notifications on a standard or custom channel, or a platform event selected on a custom channel.

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

PlatformEventChannelMember components have the suffix `.platformEventChannelMember` and are stored in the `platformEventChannelMembers` folder.

Version

PlatformEventChannelMember components are available in API version 47.0 and later.

Special Access Rules

You must have the Customize Application permission to deploy and retrieve this type.

Fields

Field Name	Field Type	Description
`enrichedFields`	EnrichedField[]	One or more fields selected for Change Data Capture Enrichment. A non-empty enriched field is added to an update or delete change event even when not changed. For more information, see Enrich Change Events with Extra Fields When Subscribed with CometD in the *Change Data Capture Developer Guide*. Available in API version 51.0 and later.
`eventChannel`	string	Required. The name of a channel. For the standard channel, the name is `ChangeEvents`. For a custom channel, the name is in this format: ***`MyChannel`***`__chn`.
`filterExpression`	string	An expression that is used to filter the stream of events and deliver only the events that match specific criteria. The filter expression can contain one or more field-value expressions. The filter expression format is based on SOQL and supports a subset of SOQL operators and field types. For example, this filter expression delivers only events that contain the City__c field with a value of 'San Francisco'. `City__c = 'San Francisco'` For more information, see Filter Your Stream of Platform Events with Custom Channels in the *Platform Events Developer Guide* and Filter Your Stream of Change Events with Channels in the *Change Data Capture Developer Guide*. Available in API version 56.0 and later.
`selectedEntity`	string	Required. The change event name of an entity selected for Change Data Capture notifications. For example, for the Account standard object, the name is `AccountChangeEvent`, or for a custom object MyObject__c, the name is `MyObject__ChangeEvent`.

EnrichedField

A field selected on PlatformEventChannelMember for Change Data Capture Enrichment. A non-empty enriched field is added to an update or delete change event even when not changed.

Field Name	Field Type	Description
`name`	string	The name of a field selected to enrich change events with.

Usage

The createMetadata() and deleteMetadata() calls aren't supported with the PlatformEventChannelMember metadata type.

To delete a channel member from a channel, deploy `destructiveChanges.xml` for this type and specify the full name of the member.

Declarative Metadata Sample Definition

This PlatformEventChannelMember component represents the selection of the Lead change event as part of the Change Data Capture selections (the standard `ChangeEvents` channel).

```
<?xml version="1.0" encoding="UTF-8"?>
<PlatformEventChannelMember xmlns="http://soap.sforce.com/2006/04/metadata">
    <eventChannel>ChangeEvents</eventChannel>
    <selectedEntity>LeadChangeEvent</selectedEntity>
</PlatformEventChannelMember>
```

Note: The file name of the example component is `ChangeEvents_LeadChangeEvent.platformEventChannelMember`. The file name, without the extension, corresponds to the component full name (`ChangeEvents_LeadChangeEvent`).

If the channel has more than one selected entity, each entity is represented separately by a PlatformEventChannelMember component. For example, this component is a second member of the standard ChangeEvents channel and represents the Contact change event.

```
<?xml version="1.0" encoding="UTF-8"?>
<PlatformEventChannelMember xmlns="http://soap.sforce.com/2006/04/metadata">
    <eventChannel>ChangeEvents</eventChannel>
    <selectedEntity>ContactChangeEvent</selectedEntity>
</PlatformEventChannelMember>
```

This example is a selected entity on the `SalesEvents__chn` custom channel.

```
<?xml version="1.0" encoding="UTF-8"?>
<PlatformEventChannelMember xmlns="http://soap.sforce.com/2006/04/metadata">
    <eventChannel>SalesEvents__chn</eventChannel>
    <selectedEntity>ContactChangeEvent</selectedEntity>
</PlatformEventChannelMember>
```

This example shows one enriched field, `Phone`, for a selected entity on the `SalesEvents__chn` custom channel. Enriched fields are supported in API version 51.0 and later.

```
<?xml version="1.0" encoding="UTF-8"?>
<PlatformEventChannelMember xmlns="http://soap.sforce.com/2006/04/metadata">
    <enrichedFields>
        <name>Phone</name>
    </enrichedFields>
    <eventChannel>SalesEvents__chn</eventChannel>
    <selectedEntity>ContactChangeEvent</selectedEntity>
</PlatformEventChannelMember>
```

This example shows a filter expression for a ContactChangeEvent selected entity on the `SalesEvents__chn` custom channel.

```
<?xml version="1.0" encoding="UTF-8"?>
<PlatformEventChannelMember xmlns="http://soap.sforce.com/2006/04/metadata">
    <eventChannel>SalesEvents__chn</eventChannel>
    <filterExpression><![CDATA[(Region__c='AMER')]]></filterExpression>
    <selectedEntity>ContactChangeEvent</selectedEntity>
</PlatformEventChannelMember>
```

Underscores in Channel Member Full Names

Two consecutive underscores in full names designate either a component name suffix or a namespace prefix. In all other cases, two consecutive underscores aren't supported in full names. If your channel member name contains a custom channel name to make it unique, ensure to replace the double underscores in the name with one underscore. For example, the member name would be `SalesEvents_chn_AccountChangeEvent` and not `SalesEvents__chn_AccountChangeEvent`.

Referencing Channel Members and Channels in `Package.xml`

This manifest file references the example definitions on the `ChangeEvents` standard channel. It lists each member in the `<members>` field of `PlatformEventChannelMember`. The `<members>` field contains the channel member full name in this format: ***`ChannelName_EventName`***.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>ChangeEvents_LeadChangeEvent</members>
        <members>ChangeEvents_ContactChangeEvent</members>
        <name>PlatformEventChannelMember</name>
    </types>
    <version>64.0</version>
</Package>
```

This manifest file references members of the `SalesEvents__chn` custom channel.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>SalesEvents_chn_AccountChangeEvent</members>
        <members>SalesEvents_chn_ContactChangeEvent</members>
        <members>SalesEvents_chn_MyCustomObj_ChangeEvent</members>
        <name>PlatformEventChannelMember</name>
    </types>
    <version>64.0</version>
</Package>
```

To retrieve a custom channel and channel members, you can reference them in the same `package.xml` file, as this example shows.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>SalesEvents__chn</members>
        <name>PlatformEventChannel</name>
    </types>
    <types>
        <members>SalesEvents_chn_AccountChangeEvent</members>
        <members>SalesEvents_chn_ContactChangeEvent</members>
        <members>SalesEvents_chn_MyCustomObj_ChangeEvent</members>
        <name>PlatformEventChannelMember</name>
    </types>
    <version>64.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SEE ALSO:

Change Data Capture Developer Guide: Example Diagrams for Channels and Channel Members

Change Data Capture Developer Guide: Filter Your Stream of Change Events with Channels

Platform Events Developer Guide: Filter Your Stream of Platform Events with Channels

PlatformEventChannel

PlatformEventSubscriberConfig

Represents configuration settings for a platform event Apex trigger, including the batch size and the trigger's running user.

This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

PlatformEventSubscriberConfig components have the suffix `.platformEventSubscriberConfig` and are stored in the `PlatformEventSubscriberConfigs` folder.

Version

PlatformEventSubscriberConfig components are available in API version 51.0 and later.

Fields

Field Name	Field Type	Description
`batchSize`	int	A custom batch size, from 1 through 2,000, for the platform event Apex trigger. The batch size corresponds to the maximum number of event messages that can be sent to a trigger in one execution. The default batch size is 2,000 for platform event triggers. We don't recommend setting the batch size to 1 to process one event at a time. Small batch sizes can slow down the processing of event messages.
`isProtected`	boolean	(Inherited field.) Indicates whether this component is protected (`true`) or not (`false`). Protected components can't be linked to or referenced by components created in a subscriber org. A developer can delete a protected component in a future release without worrying about failing

Field Name	Field Type	Description
		installations. However, once a component is marked as unprotected and is released globally, the developer can't delete it.
`masterLabel`	string	Required. The label for the PlatformEventSubscriberConfig component.
`numPartitions`	int	Specifies the number of parallel subscriptions, or partitions, that are created internally for an Apex trigger. Use this field to set up parallel subscriptions for the platform event Apex trigger. It can be an integer from 1 through 10. See Platform Event Processing at Scale with Parallel Subscriptions for Apex Triggers in the *Platform Events Developer Guide*. The default value is 1. This field is available in API version 62.0 and later.
`partitionKey`	string	Can be the standard `EventUuid` field or a required custom field of the custom platform event that the Apex trigger subscribes to. For the standard `EventUuid` field, the partition key format is the field name without the event name: `EventUuid`. For a custom field, the partition key includes the event name as a prefix in this format: ***`EventName__e.FieldName__c`***. Based on the field's generated hash value, the system determines which partition to send the event to. Use this field to specify the platform event field that is used as a partition key for parallel subscriptions. See Platform Event Processing at Scale with Parallel Subscriptions for Apex Triggers in the *Platform Events Developer Guide*. The default value is `EventUuid`. This field is available in API version 62.0 and later.
`platformEventConsumer`	string	Required. The full name of the platform event Apex trigger to configure.
`user`	string	The username of the user that the platform event Apex trigger runs as. By default, the platform event trigger runs as the Automated Process entity. Setting the running user to a specific user has these benefits: • Records are created or modified as this user. • Records with `OwnerId` fields have their `OwnerId` fields populated to this user when created or modified. • Debug logs for the trigger execution are created by this user. • You can send email from the trigger, which isn't supported with the default Automated Process user.

Declarative Metadata Sample Definition

This PlatformEventSubscriberConfig component has the label `OrderEventTriggerConfig`. It contains the configuration of a platform event trigger, `OrderEventTrigger`, and specifies the batch size and user.

```
<?xml version="1.0" encoding="UTF-8"?>
<PlatformEventSubscriberConfig xmlns="http://soap.sforce.com/2006/04/metadata">
    <platformEventConsumer>OrderEventTrigger</platformEventConsumer>
```

```
    <batchSize>200</batchSize>
    <masterLabel>OrderEventTriggerConfig</masterLabel>
    <user>user@example.com</user>
    <isProtected>false</isProtected>
</PlatformEventSubscriberConfig>
```

PlatformEventSubscriberConfig references an Apex trigger, which depends on a platform event. If the referenced items exist in the Salesforce org, you can deploy the PlatformEventSubscriberConfig component. This `package.xml` specifies the PlatformEventSubscriberConfig component.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <name>PlatformEventSubscriberConfig</name>
        <members>OrderEventTriggerConfig</members>
    </types>
    <version>64.0</version>
</Package>
```

If the referenced trigger and platform event don't exist in the org, include their definitions in the package. Otherwise, the deployment fails. This example `package.xml` includes all the referenced components.

- CustomObject represents the platform event.
- CustomField represents a custom field defined on the platform event.
- ApexTrigger represents the platform event trigger.
- PlatformEventSubscriberConfig represents the configuration options for the platform event trigger.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <name>CustomObject</name>
        <members>PlatformEvent__e</members>
    </types>
    <types>
        <name>CustomField</name>
        <members>PlatformEvent__e.Message__c</members>
    </types>
    <types>
        <name>ApexTrigger</name>
        <members>OrderEventTrigger</members>
    </types>
    <types>
        <name>PlatformEventSubscriberConfig</name>
        <members>OrderEventTriggerConfig</members>
    </types>
    <version>64.0</version>
</Package>
```

To specify all PlatformEventSubscriberConfig components, use the wildcard character, as shown in this example.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <name>PlatformEventSubscriberConfig</name>
        <members>*</members>
```

```
    </types>
    <version>64.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

Portal

The Portal metadata type represents a partner portal.

It extends Metadata and inherits its `fullName` field. To use this metadata type, you must have a partner portal or Customer Portal enabled for your organization. For more information, see Partner Portal Overview in Salesforce Help.

Declarative Metadata File Suffix and Directory Location

Lightning Platform Portal components are stored in the `portals` directory of the corresponding package directory. The file name matches the portal name, and the extension is `.portal`.

Version

Lightning Platform Portal components are available in API version 15.0 and later.

Special Access Rules

All users, including unauthenticated guest users, can view portals via the API.

Fields

Field	Field Type	Description
`active`	boolean	Required. Denotes whether this portal is active.
`admin`	string	The full name of the user designated to administer the portal.
`defaultLanguage`	string	The default language for HTML messages for the portal. Use the abbreviation for the language, for example, en_US for United States English.
`description`	string	The portal description.
`emailSenderAddress`	string	Required. The email address used when sending emails using templates configured from the portal (for example, for resetting the password).

Field	Field Type	Description
emailSenderName	string	Required. The name to display when sending emails using templates configured from the portal (for example, for resetting the password).
enableSelfCloseCase	boolean	For the Customer Portal, allows portal users to close their own cases.
footerDocument	string	The file to be used as the footer for this portal.
forgotPassTemplate	string	The email template to use when a user clicks the **Forgot Password** link. Lightning email templates aren't packageable. We recommend using a Classic email template.
fullName	string	Required. The name of the portal. Inherited from Metadata, this field is defined in the WSDL for this metadata type. It must be specified when creating, updating, or deleting. See `createMetadata()` to see an example of this field specified for a call.
headerDocument	string	The file to be used as the header for this portal.
isSelfRegistrationActivated	boolean	Determines whether self-registration is active or not for this portal.
loginHeaderDocument	string	The file to be used as the header for this portal's login page.
logoDocument	string	The file to be used as the logo for this portal.
logoutUrl	string	The URL that the user is redirected to on logout.
newCommentTemplate	string	The email template to be used for auto-notifications on new case comments.
newPassTemplate	string	The email template to be used for auto-notifications on password reset. Lightning email templates aren't packageable. We recommend using a Classic email template.
newUserTemplate	string	The email template to be used for auto-notifications on new user creation. Lightning email templates aren't packageable. We recommend using a Classic email template.
ownerNotifyTemplate	string	The email template to be used for auto-notifications on owner change. Lightning email templates aren't packageable. We recommend using a Classic email template.
selfRegNewUserUrl	string	The URL of the self-registration page.

Field	Field Type	Description
selfRegUserDefaultProfile	string	The default profile for self-registered users.
selfRegUserDefaultRole	PortalRoles (enumeration of type string)	The default role for self-registered users. The valid values are: • Executive • Manager • User • PersonAccount
selfRegUserTemplate	string	The email template to be used for auto-notifications on self-registration. Lightning email templates aren't packageable. We recommend using a Classic email template.
showActionConfirmation	boolean	Determines whether confirmation messages are displayed for actions in the portal.
stylesheetDocument	string	The Document object to be used as the CSS style sheet for this portal.
type	PortalType (enumeration of type string)	Required. The type for this portal. The valid values are: • CustomerSuccess • Partner

Declarative Metadata Sample Definition

Here's a sample XML definition of a portal.

```
<?xml version="1.0" encoding="UTF-8"?>
<Portal xmlns="http://soap.sforce.com/2006/04/metadata">
    <active>true</active>
    <description>Customer Portal</description>
    <emailSenderName>rguest@albany.com</emailSenderName>
    <enableSelfCloseCase>false</enableSelfCloseCase>
    <forgotPassTemplate>unfiled$public/ChangePwdEmail</forgotPassTemplate>
    <isSelfRegistrationActivated>false</isSelfRegistrationActivated>
    <newPassTemplate>unfiled$public/ChangePwdEmail</newPassTemplate>
    <newUserTemplate>unfiled$public/NewUserEmail</newUserTemplate>
    <selfRegUserTemplate>unfiled$public/SelfRegUserEmail</selfRegUserTemplate>
    <showActionConfirmation>false</showActionConfirmation>
    <type>CustomerSuccess</type>
</Portal>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SEE ALSO:

CustomSite

PortalDelegablePermissionSet

Represents the org-level permission sets that can be assigned to a particular profile for external users or shoppers in a store after enabling the Delegable Administration perm.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

PortalDelegablePermissionSet components have the suffix `.portaldelegablepermissionset` and are stored in the `portaldelegablepermissionsets` folder.

Version

PortalDelegablePermissionSet components are available in API version 56.0 and later.

Fields

Field Name	Description
`isProtected`	**Field Type** boolean **Description** An auto-generated value that doesn't impact the behavior of the metadata type.
`masterLabel`	**Field Type** string **Description** Required. The label for the service that appears to users.
`permissionSet`	**Field Type** string

Field Name	Description
	Description Required. Foreign key to the permissionSet on page 1568 entity.
profile	**Field Type** string **Description** Required. Foreign key to the profile on page 1616 entity.

Declarative Metadata Sample Definition

The following is the definition of the PortalDelegablePermissionSet entity.

```
<xsd:complexType name="PortalDelegablePermissionSet">
    <xsd:complexContent>
     <xsd:extension base="tns:Metadata">
      <xsd:sequence>
       <xsd:element name="isProtected" minOccurs="0" type="xsd:boolean"/>
       <xsd:element name="masterLabel" type="xsd:string"/>
       <xsd:element name="permissionSet" type="xsd:string"/>
       <xsd:element name="profile" type="xsd:string"/>
      </xsd:sequence>
     </xsd:extension>
    </xsd:complexContent>
   </xsd:complexType>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>PortalDelegablePermissionSet</name>
    </types>
    <types>
        <members>*</members>
        <name>Profile</name>
    </types>
    <types>
        <members>*</members>
        <name>PermissionSet</name>
    </types>
<version>56.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

PostTemplate

Represents the metadata associated with an approval post template for Approvals in Chatter. With approval post templates, you can customize the information included in approval request posts that appear in Chatter feeds. This type extends the Metadata metadata type and inherits its `fullName` field.

Note: Review Chatter Post Templates for Approval Requests in the Salesforce Help before you create a post template.

File Suffix and Directory Location

PostTemplate components have the suffix `.postTemplate` and are stored in the `postTemplates` folder.

Version

PostTemplate components are available in API version 29.0 and later.

Fields

Field Name	Field Type	Description
`default`	boolean	Required. Specifies whether this is the default post template for the given object. When set to `true`, this post template is used by approval processes that are associated with the same object and don't specify a post template. When an object has no default post template, each of its approval processes uses the system default post template, unless the approval process specifies its own post template.
`description`	string	Optional description of the post template.
`fields`	string[]	Required. An array of up to four fields to include in approval request posts. If the approval object is a detail object in a master-detail relationship, `Owner` isn't available for approval page layouts or approval post templates.
`label`	string	Required. Name of the post template. This non-unique label is different from the unique name of the post template.

Declarative Metadata Sample Definition

The following is an example of a PostTemplate component:

```
<PostTemplate xmlns="http://soap.sforce.com/2006/04/metadata">
    <default>false</default>
    <fields>NumberOfEmployees</fields>
    <fields>NumberofLocations__c</fields>
    <fields>PartnerAccount</fields>
    <fields>LeadCustomFieldNumber__c</fields>
```

```
    <label>My Lead Post Template</label>
</PostTemplate>
```

The following is an example package manifest that references the previous PostTemplate component.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>Lead.leadtemplate</members>
        <name>PostTemplate</name>
    </types>
    <version>29.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ProductAttributeSet

Represents the ProductAttribute information being used as and attribute such as color_*c*, *size*_c .

Version

ProductAttributeSet components are available in API version 54 and later.

Special Access Rules

Fields

Field Name	Field Type	Description
`description`	string	A meaningful explanation of the attribute set.
`developerName`	string	A unique name for the attribute set.
`masterLabel`	string	The name of the attribute set.
`productAttributeSetItems`	ProductAttributeSetItem	A list of ProductAttributeSetItem.

PresenceDeclineReason

Represents an Omni-Channel decline reason that agents can select when declining work requests. This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

PresenceDeclineReason components have the suffix `.presenceDeclineReason` and are stored in the `presenceDeclineReasons` folder.

Version

PresenceDeclineReason components are available in API version 44.0 and later.

Special Access Rules

This type is available only if Omni-Channel is enabled in your org.

Fields

Field Name	Field Type	Description
`label`	string	The label for the decline reason.

Declarative Metadata Sample Definition

The following is an example of a PresenceDeclineReason component.

```
<?xml version="1.0" encoding="UTF-8"?>
<PresenceDeclineReason xmlns="http://soap.sforce.com/2006/04/metadata">
    <label>Incorrect queue</label>
</PresenceDeclineReason>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>PresenceDeclineReason</name>
    </types>
    <version>44.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

PresenceUserConfig

Represents a configuration that determines a presence user's settings.

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

PresenceUserConfig components have the suffix `.presenceUserConfig` and are stored in the `presenceUserConfigs` folder.

Version

PresenceUserConfig components are available in API version 44.0 and later.

Special Access Rules

This type is available only if Omni-Channel is enabled in your org.

Fields

Field Name	Field Type	Description
`assignments`	PresenceConfigAssignments	Specifies how presence configurations are assigned to Omni-Channel users. Presence configurations can be assigned to sets of users or to sets of profiles.
`capacity`	int	Required. The maximum number of work units an agent can be assigned at one time.
`declineReasons`	string	Specifies the list of decline reasons that an agent can select when they decline a work.
`enableAutoAccept`	boolean	Indicates whether work items that are routed to agents are automatically accepted (`true`) or not (`false`). Available only if `enableDecline` is set to `false`.
`enableDecline`	boolean	Indicates whether agents can decline work items that are routed to them (`true`) or not (`false`). Available only if `enableAutoAccept` is set to `false`.
`enableDeclineReason`	boolean	Indicates whether agents can select a reason for declining work requests (`true`) or not (`false`). This can be selected only if decline reasons are enabled.
`enableDisconnectSound`	boolean	Indicates whether a sound is played when agents are disconnected from Omni-Channel (`true`) or not (`false`).
`enableRequestSound`	boolean	Indicates whether a sound plays with incoming work requests (`true`) or not (`false`). Set to `true` by default.
`interruptibleCapacity`	int	Indicates the maximum number of work units using interruptible capacity that can be pushed to an agent at a time. An empty value defaults this field to the value set in the `capacity` field. Available in API version 57.0 and later when the Interruptible Capacity feature is enabled.
`label`	string	The label of the presence configuration.

Field Name	Field Type	Description
presenceStatusOnDecline	string	The presence status that's automatically assigned to the agent when the agent declines a work item. Available only if enableDecline is set to true.
presenceStatusOnPushTimeout	string	The presence status that's automatically assigned to the agent when the agent doesn't respond to a work item before push timeout occurs.

PresenceConfigAssignments

Represents the assignments of an org's profiles and users to a Presence configuration.

Field Name	Field Type	Description
profiles	PresenceConfigProfileAssignments	Specifies the profiles that are associated with a specific presence configuration.
users	PresenceConfigUserAssignments	Specifies the users that are associated with a specific presence configuration.

PresenceConfigProfileAssignments

Represents the profiles associated with a specific presence configuration.

Field Name	Field Type	Description
profile	string	Specifies the name of the profile associated with a specific presence configuration.

PresenceConfigUserAssignments

Represents the users associated with a specific presence configuration.

Field Name	Field Type	Description
user	string	Specifies the username of the user associated with a specific presence configuration.

Declarative Metadata Sample Definition

The following is an example of a PresenceUserConfig component.

```
<?xml version="1.0" encoding="UTF-8"?>
<PresenceUserConfig xmlns="http://soap.sforce.com/2006/04/metadata">
    <assignments>
        <profiles>
            <profile>standard</profile>
        </profiles>
        <users>
            <user>jdoe@example.com</user>
```

```
        </users>
    </assignments>
    <capacity>5</capacity>
    <declineReasons>Incorrect_queue</declineReasons>
    <enableAutoAccept>false</enableAutoAccept>
    <enableDecline>true</enableDecline>
    <enableDeclineReason>true</enableDeclineReason>
    <enableDisconnectSound>true</enableDisconnectSound>
    <enableRequestSound>true</enableRequestSound>
    <label>My presence configuration</label>
    <presenceStatusOnDecline>Away</presenceStatusOnDecline>
    <presenceStatusOnPushTimeout>Break</presenceStatusOnPushTimeout>
</PresenceUserConfig>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>PresenceUserConfig</name>
    </types>
    <version>44.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

PricingActionParameters

Represents the pricing action that's associated with a context definition and pricing procedure.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

PricingActionParameters components have the suffix `.pricingActionParameters` and are stored in the `pricingActionParameters` folder.

Version

PricingActionParameters components are available in API version 60.0 and later.

Special Access Rules

This metadata type is available with Salesforce Pricing.

Fields

Field Name	Description
`contextDefinition`	**Field Type** string **Description** Required. Context definition record that's associated with the pricing action.
`contextMapping`	**Field Type** string **Description** Required. Context mapping record that's associated with the pricing action.
`developerName`	**Field Type** string **Description** Required. Unique name of the pricing action parameter record. The name must begin with a letter and use only alphanumeric characters and underscores. The name must not include spaces, end with an underscore, or have two consecutive underscores.
`effectiveFrom`	**Field Type** dateTime **Description** Required. Date and time from when the pricing action becomes effective.
`effectiveTo`	**Field Type** dateTime **Description** Date and time till when the pricing action is in effect.
`masterLabel`	**Field Type** string

Field Name	Description
	Description Required. Master label of the pricing action parameter.
`objectName`	**Field Type** string **Description** Name of the object that's associated with the pricing action. Valid values are: • `Case` • `Contract` • `Opportunity` • `Order` • `Quote` • `SalesAgreement` • `WorkOrder`
`pricingProcedure`	**Field Type** string **Description** Pricing procedure record that's associated with this pricing action.

Declarative Metadata Sample Definition

The following is an example of a PricingActionParameters component.

```
<PricingActionParameters xmlns="http://soap.sforce.com/2006/04/metadata">
    <developerName>CMEDefaultActionParameters</developerName>
    <objectName>ORDER</objectName>
    <pricingProcedure>PP</pricingProcedure>
    <effectiveFrom>2024-04-08T07:32:00.000Z</effectiveFrom>
    <effectiveTo>2024-04-11T07:32:00.000Z</effectiveTo>
    <contextDefinition>SalesTransactionContext__stdctx</contextDefinition>
    <contextMapping>SalesAgreementEntitiesMapping</contextMapping>
    <masterLabel>PAP_test</masterLabel>
</PricingActionParameters>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>PricingActionParameters</name>
    </types>
```

```
    <version>64.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

PricingRecipe

Represents the data models or sets of objects of a particular cloud that the pricing data store consumes during design time and run time.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

PricingRecipe components have the suffix `.pricingRecipe` and are stored in the `pricingRecipe` folder.

Version

PricingRecipe components are available in API version 60.0 and later.

Special Access Rules

This metadata type is available with Salesforce Pricing.

Fields

Field Name	Description
`defaultPricingProcedure`	**Field Type** ExpressionSetDefinition **Description** Expression set definition that's associated with this pricing recipe setting.
`defaultPricingProcedureDeveloperName`	**Field Type** string **Description** For internal use only.

Field Name	Description
`defaultPricingProcedureId`	**Field Type** string **Description** ID of the pricing procedure of the pricing recipe.
`developerName`	**Field Type** string **Description** Required. API name of the pricing recipe.
`isActive`	**Field Type** boolean **Description** Indicates whether the pricing recipe is active (`true`) or not (`false`). The default value is `false`
`isInternal`	**Field Type** boolean **Description** Indicates whether the price recipe record is created internally by the Salesforce platform (`true`) or not (`false`). The default value is `false`
`masterLabel`	**Field Type** string **Description** Required. Name for pricing recipe that's defined when the pricing recipe is created.
`pricingRecipeTableMapping`	**Field Type** PricingRecipeTableMapping[] **Description** Mapping of the pricing components of a lookup table with the chosen pricing recipe.

PricingRecipeTableMapping

Represents the mapping of the lookup table with the chosen pricing recipe.

Field Name	Description
`isInternal`	**Field Type** boolean **Description** Indicates whether the price recipe field mapping record is created internally by the Salesforce platform (`true`) or not (`false`). The default value is `false`.
`lookupTable`	**Field Type** DecisionTable DecisionMatrixDefinition **Description** Lookup table that's associated with either a decision matrix or decision table.
`lookupTableDeveloperName`	**Field Type** string **Description** For internal use only.
`pricingComponentType`	**Field Type** string **Description** Pricing component field data that the decision table is built on. Valid values are: • `AttributeDiscount` • `BundleDiscount` • `DerivedPricing` • `ListPrice` • `PriceAdjustmentMatrix` • `PromotionsDiscount` • `VolumeDiscount` • `VolumeTierDiscount` • `DiscountDistributionService`. This value is available in API version 60.0 and later. • `MinimumPrice`. Available in API version 62.0 and later.
`pricingProcedureOutputMapList`	**Field Type** PricingProcedureOutputMap[] **Description** List of the mappings of the outputs of the pricing procedures to the associated lookup tables. Available in API version 60.0 and later.

Field Name	Description
`pricingRecipe`	**Field Type** string **Description** Required. Pricing data store that's associated with this pricing recipe field mapping.

PricingProcedureOutputMap

Represents the mapping of the outputs of the pricing procedures to the associated lookup tables. Each record specifies the output mapping of the associated lookup table based on the pricing component type specified in the PricingRecipeTableMapping object.

Field Name	Description
`fieldName`	**Field Type** string **Description** For internal use only.
`isPricingRecipeActive`	**Field Type** boolean **Description** Indicates whether the associated pricing recipe is active (`true`) or not (`false`). The default value is `false`.
`outputFieldName`	**Field Type** string **Description** Field name that contains the output type that's generated from the pricing element.
`outputFieldNameString`	**Field Type** string **Description** Derived field that references a specific column in a decision table or decision matrix.
`outputType`	**Field Type** string **Description** Output type that's generated from a pricing element. Valid values are: • `AdjustmentType` • `AdjustmentValue`

Field Name	Description
	• CustomOutput • HashOutput • UnitPrice

Declarative Metadata Sample Definition

The following is an example of a PricingRecipe component.

```
<PricingRecipe xmlns="http://soap.sforce.com/2006/04/metadata">
    <defaultPricingProcedureId> </defaultPricingProcedureId>
    <developerName>CMEDefaultRecipe</developerName>
    <isActive>false</isActive>
    <isInternal>false</isInternal>
    <masterLabel>CMEDefaultRecipe</masterLabel>
    <pricingRecipeTableMapping>
        <isInternal>false</isInternal>

<lookupTableDeveloperName>Bundle_Based_Adjustment_Decision_Table</lookupTableDeveloperName>

        <pricingComponentType>CUSTOMDISCOUNT</pricingComponentType>
        <fileBasedDecisionTableName>Bundle Based Adjustment
Entries</fileBasedDecisionTableName>
        <pricingProcedureOutputMapList>
            <fieldName>AdjustmentValue</fieldName>
            <isPricingRecipeActive>false</isPricingRecipeActive>
            <outputFieldName>01Pxx000000000f</outputFieldName>
            <outputFieldNameString>false</outputFieldNameString>
            <outputType>AdjustmentValue</outputType>
        </pricingProcedureOutputMapList>
        <pricingProcedureOutputMapList>
            <fieldName>AdjustmentType</fieldName>
            <isPricingRecipeActive>false</isPricingRecipeActive>
            <outputFieldName>01Pxx000000000m</outputFieldName>
            <outputFieldNameString>false</outputFieldNameString>
            <outputType>AdjustmentType</outputType>
        </pricingProcedureOutputMapList>
        <pricingRecipe>CMEDefaultRecipe</pricingRecipe>
    </pricingRecipeTableMapping>
</PricingRecipe>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>PricingRecipe</name>
    </types>
    <version>64.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

Profile

Represents a user profile. A profile defines a user's permission to perform different functions within Salesforce. This type extends the Metadata metadata type and inherits its `fullName` field.

In API version 29.0 and later, you can retrieve and deploy access settings for these managed components in profiles and permission sets:

- Apex classes
- Apps
- Custom field permissions
- Custom object permissions
- Custom tab settings
- External data sources
- Record types
- Visualforce pages

In API version 51.0 and later, you can retrieve and deploy access settings for login flows. For more information, see Managed Component Access in the Components in a Module section of Sample `package.xml` Manifest Files.

As of API version 50.0 and later, only users with correct permissions can view profile names other than their own if the Profile Filtering setting is enabled.

Important: Profile names are also exposed when users with permissions to perform the following tasks take these actions:

- Create a tab or record type with a wizard step that includes the assignment of tabs and record types to profiles.
- Configure a login flow where viewing profile lists is required to make flow associations.
- Set up delegated admins where looking up profiles is needed to identify assignable profiles.
- Administer an org as a delegated customer admin.
- Administer an org as a delegated admin to view and assign profiles of the delegated group.

Declarative Metadata File Suffix and Directory Location

The file suffix is `.profile`. There's one file for each profile, stored in the `profiles` folder in the corresponding package directory.

Version

Profiles are available in API version 10.0 and later.

Special Access Rules

As of Summer '20 and later, Customer Portal and Partner Portal users can't access this type.

To view the following settings, assignments, and permissions for standard and custom objects in a specified profile, the View Setup and Configuration permission is required.

- Client settings
- Field permissions
- Layout assignments
- Object permissions
- Permission dependencies
- Permission set tab settings
- Permission set group components
- Record types

Fields

The content of a profile returned by Metadata API depends on the content requested in the `RetrieveRequest` message. For example, profiles only include field-level security for fields included in custom objects returned in the same `RetrieveRequest` as the profiles. The profile definition contains the following fields:

Important: We designed Profile metadata deployment to overlay the existing Profile settings in a target org. For example, if you disable permissions for a profile, the newly disabled permission information isn't exported. To force all Profile changes to deploy through metadata, including permission disablement, add code that explicitly indicates disabled permissions. For example, add this code to the Profile metadata `.xml` file before deploying into a target org: `<value>false</value>`.

If you deploy a profile that doesn't exist in the target org and don't specify any permissions or settings, then the resulting profile contains all permissions and settings in the standard Minimum Access - Salesforce profile (API version 60.0 and later) or the standard Standard User profile (API version 59.0 and earlier).

Note: As of API version 38.0, you can change field permissions to make a field editable using the Metadata API for fields that you can't change through the user interface. For example, you can deploy `Asset.ProductCode` as an editable field even though you can't through the user interface.

Field Name	Field Type	Description
`agentAccesses`	ProfileAgentAccess[]	Indicates which agents are visible to users assigned to this profile. Available in API version 63.0 and later.
`applicationVisibilities`	ProfileApplicationVisibility[]	Indicates which apps are visible to users assigned to this profile. In API version 29.0 and earlier, this field supports custom apps only. In API version 30.0 and later, this field supports both standard and custom apps.
`categoryGroupVisibilities`	ProfileCategoryGroupVisibility[]	Indicates which data category groups are visible to users assigned to this profile. Available in API version 41.0 and later.
`classAccesses`	ProfileApexClassAccess[]	Indicates which top-level Apex classes have methods that users assigned to this profile can execute.
`custom`	boolean	Indicates whether the profile is a custom (`true`) or standard (`false`) profile. Available in API version 30.0 and later.

Field Name	Field Type	Description
customMetadataTypeAccesses	ProfileCustomMetadataTypeAccess[]	Indicates the custom metadata types that are read-accessible to a user assigned to this profile. Available in API version 47.0 and later.
customPermissions	ProfileCustomPermissions[]	Indicates which custom permissions are available to users assigned to this profile. Available in API version 31.0 and later.
customSettingAccesses	ProfileCustomSettingAccesses[]	Indicates the custom settings that are read-accessible to a user assigned to this profile. Available in API version 47.0 and later.
description	string	The profile description. Limit: 255 characters. Available in API version 30.0 and later.
externalDataSourceAccesses	ProfileExternalDataSourceAccess[]	Indicates which data sources with identity type of `Per User` are available to users assigned to this profile. Available in API version 27.0 and later.
fieldLevelSecurities	ProfileFieldLevelSecurity[]	Indicates which fields are visible to a user assigned to this profile, and the kind of access available (editable or hidden). This field is available in API version 22.0 and earlier.
fieldPermissions	ProfileFieldLevelSecurity[]	Indicates which fields are visible to a user assigned to this profile, and the kind of access available (editable or readable). This field is available in API version 23.0 and later.
flowAccesses	ProfileFlowAccess[]	Indicates which flows can be accessed by a user assigned to this profile. Available in API version 47.0 and later.
fullName	string	The name can only contain characters, letters, and the underscore (_) character. The name must start with a letter, and can't end with an underscore or contain two consecutive underscore characters. Inherited from the Metadata component, this field isn't defined in the WSDL for this component. It must be specified when creating, updating, or deleting. See create() to see an example of this field specified for a call.
layoutAssignments	ProfileLayoutAssignments[]	Indicates which layout to use for this profile.
loginFlows	LoginFlow[]	Indicates a business process that you direct users to before they access Salesforce.
loginHours	ProfileLoginHours[]	Indicates the hours within which a user with this profile can log in. If not specified, the profile doesn't restrict a user's login hours. This field is available in API version 25.0 and later.

Field Name	Field Type	Description
`loginIpRanges`	ProfileLoginIpRange[]	The list of IP address ranges from which users with a particular profile can log in. This field is available in API version 17.0 and later.
`objectPermissions`	ProfileObjectPermissions[]	Indicates which objects are accessible to a user assigned to this profile, and the kind of access available (create, read, edit, delete, and so on). In API version 28.0 and later, this field is only retrieved when `allowRead` is `true`. In API version 50.0 and later, editing standard objects on standard profiles is disabled.
`pageAccesses`	ProfileApexPageAccess[]	Indicates which Visualforce pages that users assigned to this profile can execute.
`profileActionOverrides`	ProfileActionOverride[]	A list of the Lightning Experience Home page action overrides that are assigned to this profile. When a user logs in with a profile, a matching ProfileActionOverride assignment takes precedence over existing overrides for the Home tab specified in ActionOverride. This field is available in API versions 37.0 to 44.0.
`recordTypeVisibilities`	ProfileRecordTypeVisibility[]	Indicates the visibility of record types for users assigned to this profile. In API version 29.0 and later, this field isn't retrieved or deployed for inactive record types.
`ServicePresenceStatusAccesses`	ProfileServicePresenceStatusAccess[] on page 1628	Indicates which Service presence statuses that the user assigned to this profile can execute. Available in API version 64.0 and later.
`tabVisibilities`	ProfileTabVisibility[]	Indicates which record types are visible to a user assigned to this profile, and therefore which tabs within an app are visible.
`userLicense`	string	The `User License` for the profile. A user license determines the baseline of features that the user can access. Every user must have exactly one user license. This field is available in API version 17.0 and later.
`userPermissions`	ProfileUserPermission[]	Specifies a user permission (such as "API Enabled") and whether it's enabled for this profile. This field retrieves only enabled user permissions. Available in API version 29.0 and later.

LoginFlow

LoginFlow represents a business process that you direct users to before they access Salesforce. You can use Metadata API to define existing flows as login flows and to edit login flow definitions. To delete login flow definitions, use the Login Flow page.

Field Name	Field Type	Description
`flow`	string	Required only if the `uiLoginFlowType` is `VisualWorkflow`. The `fullName` of the Flow. Before you can deploy the LoginFlow, the Flow referenced here must be deployed in your org and its status must be `Active`.
`flowtype`	LoginFlowType (enumeration of type string)	Required. The value is `UI`.
`friendlyname`	string	Required. The name of the LoginFlow.
`uiLoginFlowType`	UiLoginFlowType (enumeration of type string)	Required. The type of login flow. These are valid values. • `VisualWorkflow`—Indicates a Salesforce Flow. You can create these flows using Flow Builder. • `VisualForce`—Indicates a flow created using Visualforce.
`useLightningRuntime`	boolean	Indicates if Lightning Runtime is used (`true`) or not (`false` (default)). Used only if `uiLoginFlowType` is `VisualWorkflow`.
`vfFlowPage`	string	Required only if the `uiLoginFlowType` is `VisualForce`. The name of the VisualForce page.
`vfFlowPageTitle`	string	Required only if the `uiLoginFlowType` is `VisualForce`. The name of the VisualForce page.

ProfileActionOverride

ProfileActionOverride represents a user profile-based override of an ActionOverride on a standard Home tab in Lightning Experience.

Note:

- ProfileActionOverride can be defined only on Profile for API version 39.0 to 44.0. In API version 45.0 and later, ProfileActionOverride must be defined for CustomApplication instead. Beginning with API version 45.0, Home page assignments related to user profile must also have a corresponding app assignment because more granular Home page assignments are supported. As a result, ProfileActionOverride is defined for CustomApplication rather than Profile.
- ProfileActionOverride settings aren't retrieved in the `.profile` file unless a Lightning page is referenced in the `package.xml` file.

Field Name	Field Type	Description
`actionName`	string	Required. The possible values are the same as the actions you can override: • `accept` • `clone` • `delete` • `edit` • `list`

Field Name	Field Type	Description
		• `new` • `tab` • `view`
`content`	string	Set this field if `type` is set to `flexipage`, `lightningcomponent`, `scontrol`, or `visualforce`. It refers to the name of the Lightning page, Lightning component, s-control, or Visualforce page to use as the override. To reference installed components, use this format: ***`Component_namespace__Component_name`***.
`formFactor`	FormFactor (enumeration of type string)	The size of the page being overridden. The `Large` value represents the Lightning Experience desktop environment and is valid only for the `flexipage` and `lightningcomponent` types. The `Small` value represents the Salesforce mobile app on a phone or tablet. The `Medium` value is reserved for future use. The `null` value (which is the same as specifying no value) represents Salesforce Classic.
`pageOrSobjectType`	string	The name of the sObject type being overridden. Valid values are `standard` and `custom`. This value must be `standard-home` when actionName is `tab`.
`recordType`	string	The record type assigned to the ProfileActionOverride. If the `PageOrSobjectType` is `standard-home`, this field is null.
`type`	ActionOverrideType (enumeration of type string)	Required. Represents the type of action override. Valid values are described in ActionOverrideType.

ProfileAgentAccess

ProfileAgentAccess represents the agent access configuration for users assigned through a profile.

Field Name	Field Type	Description
`agentName`	string	Required. The name of the employee agent.
`enabled`	boolean	Required. Indicates whether users assigned to this profile can use the Agentforce Employee Agent (`true`) or not (`false`).

ProfileApplicationVisibility

ProfileApplicationVisibility determines whether an app is visible to a user assigned to this profile.

Field Name	Field Type	Description
`application`	string	Required. The name of the app.
`default`	boolean	Required. Indicates whether the app is the default app (`true`) or not (`false`). Only one app per profile can be set to `true`.
`visible`	boolean	Required. Indicates whether this app is visible to users assigned to this profile (`true`) or not (`false`).

ProfileCategoryGroupVisibility

ProfileCategoryGroupVisibility determines whether a data category group is visible to a user assigned to this profile. Available in API version 41.0 and later.

Field Name	Field Type	Description
`dataCategories`	string[]	Array of one or more data category names.
`dataCategoryGroup`	string	Required. The name of the data category group.
`visibility`	CategoryGroupVisibility (enumeration of type string)	Required. Indicates the visibility of the data category. Valid values are: • `ALL` • `CUSTOM` • `NONE`

ProfileCustomMetadataTypeAccess

ProfileCustomMetadataTypeAccess represents the custom metadata type access for users assigned to a profile. Available in API version 47.0 and later.

Field	Field Type	Description
`enabled`	boolean	Required. Indicates whether the records for this custom metadata type are readable (`true`) or not (`false`).
`name`	string	Required. The custom metadata type name.

ProfileApexClassAccess

ProfileApexClassAccess determines which top-level Apex classes have methods that users assigned to this profile can execute.

Field Name	Field Type	Description
`apexClass`	string	Required. The Apex class name.

Field Name	Field Type	Description
`enabled`	boolean	Required. Indicates whether users assigned to this profile can execute methods in the top-level class (`true`) or not (`false`).

ProfileCustomPermissions

ProfileCustomPermissions represents the custom permissions access for users assigned to a profile. Only enabled custom permissions are retrieved.

Field Name	Field Type	Description
`enabled`	boolean	Required. Indicates whether the custom permission is enabled (`true`) or not (`false`).
`name`	string	Required. The custom permission name.

ProfileCustomSettingAccesses

ProfileCustomSettingAccesses represents the custom setting access for users assigned to a profile. Available in API version 47.0 and later.

Field	Field Type	Description
`enabled`	boolean	Required. Indicates whether the records for this custom setting are readable (`true`) or not (`false`).
`name`	string	Required. The custom setting name.

ProfileExternalDataSourceAccess

ProfileExternalDataSourceAccess represents the data source access for users with identity type of `Per User`. Available in API version 27.0 and later.

Field Name	Field Type	Description
`enabled`	boolean	Required. Indicates whether the data source is enabled (`true`) or not (`false`).
`externalDataSource`	string	The name of the external data source.

ProfileFieldLevelSecurity

ProfileFieldLevelSecurity represents the field level security for users assigned to a profile. In API version 30.0 and later, permissions for required fields can't be retrieved or deployed.

Field Name	Field Type	Description
`editable`	boolean	Required. Indicates whether this field is editable (`true`) or not (`false`). In API version 30.0 and later, when deploying a new custom field, this field is `false` by default.
`field`	string	Required. Indicates the name of the field. When referencing shared Activity fields, specify Event or Task. For example, `Event.Meeting__c`.
`hidden`	boolean	Indicates whether this field is hidden (`true`) or not (`false`). This field is available in API version 22.0 and earlier. For portal profiles, this field is set to `true` by default in API version 19.0 and later.
`readable`	boolean	Indicates whether this field is readable (`true`) or not (`false`). This field is available in API version 23.0 and later. It replaces the `hidden` field. In API version 30.0 and later, when deploying a new custom field, this field is `false` by default. For portal profiles, this field is set to `false` by default.

ProfileFlowAccess

ProfileFlowAccess represents which flows a profile grants access to. Available in API version 47.0 and later.

Field	Field Type	Description
`enabled`	boolean	Required. Indicates whether users assigned this profile can access the flow (`true`) or not (`false`). The default value is `false`.
`flow`	string	Required. The name of the flow to which access is granted.

ProfileLayoutAssignments

ProfileLayoutAssignments determines which layout to use for a profile and a given entity.

Field Name	Field Type	Description
`layout`	string	Required. Indicates the layout for this particular entity.
`recordType`	string	This field is optional. If the `recordType` of the record matches a layout assignment rule, it uses the specified layout.

ProfileLoginHours

ProfileLoginHours restricts the days and times within which users with a particular profile can log in.

Field Name	Field Type	Description
weekday`Start`	string	Specifies the earliest time on that day that a user with this profile can log in. If a start time for a particular day is specified, an end time for that day also must be specified. Start can't be greater than end for a particular day. • Valid values for `weekday`: `monday`, `tuesday`, `wednesday`, `thursday`, `friday`, `saturday`, or `sunday`. For example, `mondayStart` indicates the beginning of the login period for Monday. • Valid values for Start: the number of minutes since midnight. Must be evenly divisible by 60 (full hours). For example, `300` is 5:00 AM.
weekday`End`	string	Specifies the time on that day that a user with this profile must log out by. • Valid values for `weekday`: `monday`, `tuesday`, `wednesday`, `thursday`, `friday`, `saturday`, or `sunday`. For example, `mondayEnd` indicates the close of the login period for Monday. • Valid values for End: the number of minutes since midnight. Must be evenly divisible by 60 (full hours). For example, `1020` is 5:00 PM.

To delete login hour restrictions from a profile that previously had them, you must explicitly include an empty loginHours tag without any start or end times.

ProfileLoginIpRange

ProfileLoginIpRange IP defines an IP address range that users with a particular profile can log in from.

Field Name	Field Type	Description
`description`	string	Use this field to identify the purpose of the range, such as which part of a network corresponds to this range. This field is available in API version 31.0 and later.
`endAddress`	string	Required. The end IP address for the range.
`startAddress`	string	Required. The start IP address for the range.

ProfileObjectPermissions

ProfileObjectPermissions represents a user's access to objects.

Note:

- In API version 18.0 and later, these permissions are disabled in new custom objects for any profiles where "View All Data" or "Modify All Data" is disabled.
- In API version 50.0 and later, editing standard objects on standard profiles is disabled.

Field Name	Field Type	Description
allowCreate	boolean	Indicates whether the object referenced by the object field can be created by the users assigned to this profile (true) or not (false). This field is named revokeCreate before version 14.0 and the logic is reversed. The field name change and the update from true to false and the reverse is automatically handled between versions and doesn't require any manual editing of existing XML component files.
allowDelete	boolean	Indicates whether the object referenced by the object field can be deleted by the users assigned to this profile (true) or not (false). This field is named revokeDelete before version 14.0 and the logic is reversed. The field name change and the update from true to false and the reverse is automatically handled between versions and doesn't require any manual editing of existing XML component files.
allowEdit	boolean	Indicates whether the object referenced by the object field can be edited by the users assigned to this profile (true) or not (false). This field is named revokeEdit before version 14.0 and the logic is reversed. The field name change and the update from true to false and the reverse is automatically handled between versions and doesn't require any manual editing of existing XML component files.
allowRead	boolean	Indicates whether the object referenced by the object field can be seen by the users assigned to this profile (true) or not (false). This field is named revokeRead before version 14.0 and the logic is reversed. The field name change and the update from true to false and the reverse is automatically handled between versions and doesn't require any manual editing of existing XML component files.
modifyAllRecords	boolean	Indicates whether all records for the object referenced by the object field can be read, edited, or deleted by the users assigned to this profile (true) or not (false), regardless of the sharing settings for the object. This setting is equivalent to the Modify All Data user permission limited to the individual object level. Available in API version 15.0 and later. This field isn't available for all objects. Refer to the profile in the user interface to determine which objects currently support these permissions. Profiles with Modify All Data ignore modifyAllRecords entries in Metadata API and don't return an error if Modify All Data is enabled on the profile.
object	string	Required. The name of the object whose permissions are altered by this profile, for example, MyCustomObject__c.
viewAllFields	boolean	Indicates whether all fields and field data for the object referenced by the object field can be read by the users assigned to this profile (true) or not (false). Available in API version 63.0 and later.

Field Name	Field Type	Description
`viewAllRecords`	boolean	Indicates whether all records for the object referenced by the `object` field can be read by the users assigned to this profile (`true`) or not (`false`), regardless of the sharing settings for the object. This setting includes private records (records with no parent object). This setting is equivalent to the View All Data user permission limited to the individual object level. Available in API version 15.0 and later. This field isn't available for all objects. Refer to the profile in the user interface to determine which objects currently support these permissions. Profiles with "View All Data" ignore `viewAllRecords` entries in the Metadata API and don't return an error if View All Data is enabled on the profile.

ProfileApexPageAccess

ProfileApexPageAccess determines which Visualforce pages that users assigned to this profile can execute.

Field Name	Field Type	Description
`apexPage`	string	Required. The Visualforce page name.
`enabled`	boolean	Required. Indicates whether users assigned to this profile can execute the Visualforce page (`true`) or not (`false`).

ProfileRecordTypeVisibility

ProfileRecordTypeVisibility represents the visibility of record types for this profile. Record types let you offer different business processes, picklist values, and page layouts to different users.

Field Name	Field Type	Description
`default`	boolean	Required. Indicates whether the record type is the default when users with this profile create records for this object (`true`) or not (`false`).
`personAccountDefault`	boolean	When Person Accounts is enabled, this field indicates whether the record type is this profile's default person account record type (`true`) or not (`false`). When Person Accounts is disabled, this field's value has no impact. Person accounts aren't enabled by default in Salesforce. To request person accounts, contact Salesforce.
`recordType`	string	Required. The record type name, for example `Account.MyRecordType`.
`visible`	boolean	Required. Indicates whether this record type is visible to users assigned to this profile (`true`) or not (`false`).

ProfileTabVisibility

ProfileTabVisibility represents the visibility of tabs for this profile. For version 17.0 and later, ProfileTabVisibility supports visibility of tabs for standard objects. The manifest file must include the standard object corresponding to a standard tab to retrieve the tab visibility in a profile.

Field Name	Field Type	Description
`tab`	string	Required. The name of the tab.
`visibility`	TabVisibility (enumeration of type string)	Required. Indicates the visibility of the tab. Valid values are: • `DefaultOff`—The tab is available on the All Tabs page. Users can individually customize their display to make the tab visible in any app. • `DefaultOn`—The tab is available on the All Tabs page and appears in the visible tabs for its associated app. Users can individually customize their display to hide the tab or make it visible in other apps. • `Hidden`—The tab isn't available on the All Tabs page or visible in any apps. In API version 36.0 and earlier, `Hidden` is returned only if `visibility` was set using the API. If it was set to `Hidden` from the profile in Salesforce, the API doesn't return a visibility value. For version 37.0 and later, when tab visibility is set to hidden, the API returns `Hidden`, regardless of how the value was set.

ProfileUserPermission

ProfileUserPermission represents an app or system permission for a profile. Use one of these elements for each permission.

Field	Field Type	Description
`enabled`	boolean	Required. Indicates whether the permission is enabled (`true`) or disabled (`false`).
`name`	string	Required. The permission name.

ProfileServicePresenceStatusAccess

Represents the presence statuses that reps assigned to this profile have access. Available in API version 64.0 and later.

Field	Field Type	Description
`servicePresenceStatus`	string	Required. The name of Service Presence Status.
`enabled`	boolean	Required. Indicates whether the rep assigned to this profile has access to the presence status (`true`) or not (`false`).

Java Sample

This sample uses picklists, profiles, record types, and a custom app:

```
public void profileSample() {
  try {
    // Create an expense report record, tab and app...
    CustomObject expenseRecord = new CustomObject();
    expenseRecord.setFullName("ExpenseReport__c");
    expenseRecord.setLabel("Expense Report");
    expenseRecord.setPluralLabel("Expense Reports");

    expenseRecord.setDeploymentStatus(DeploymentStatus.Deployed);
    expenseRecord.setSharingModel(SharingModel.ReadWrite);

    CustomField nameField = new CustomField();
    nameField.setType(FieldType.AutoNumber);
    nameField.setLabel("Expense Report Number");
    nameField.setDisplayFormat("ER-{0000}");
    expenseRecord.setNameField(nameField);

    AsyncResult[] arsExpenseRecord =
        metadataConnection.create(new Metadata[] {expenseRecord});

    Picklist expenseStatus = new Picklist();
    PicklistValue unsubmitted = new PicklistValue();
    unsubmitted.setFullName("Unsubmitted");
    PicklistValue submitted = new PicklistValue();
    submitted.setFullName("Submitted");
    PicklistValue approved = new PicklistValue();
    approved.setFullName("Approved");
    PicklistValue rejected = new PicklistValue();
    rejected.setFullName("Rejected");
    expenseStatus.setPicklistValues(new PicklistValue[] {
        unsubmitted, submitted, approved, rejected}
    );

    CustomField expenseStatusField = new CustomField();
    expenseStatusField.setFullName(
        "ExpenseReport__c.ExpenseStatus__c"
    );
    expenseStatusField.setLabel("Expense Report Status");
    expenseStatusField.setType(FieldType.Picklist);
    expenseStatusField.setPicklist(expenseStatus);
    AsyncResult[] arsStatusField =
        metadataConnection.create(new Metadata[]
            {expenseStatusField});

    CustomTab expenseTab = new CustomTab();
    expenseTab.setFullName("ExpenseReport__c");
    expenseTab.setMotif("Custom70: Handsaw");
    expenseTab.setCustomObject(true);
    AsyncResult[] arsTab =
        metadataConnection.create(new Metadata[] {expenseTab});
```

```
    CustomApplication application = new CustomApplication();
    application.setFullName("ExpenseForce");
    application.setTab(new String[] {expenseTab.getFullName()});
    AsyncResult[] arsApp =
        metadataConnection.create(new Metadata[] {application});

    // Employees and managers have the same app visibility...
    ProfileApplicationVisibility appVisibility =
        new ProfileApplicationVisibility();
    appVisibility.setApplication("ExpenseForce");
    appVisibility.setVisible(true);

    Profile employee = new Profile();
    employee.setFullName("Employee");
    employee.setApplicationVisibilities(
        new ProfileApplicationVisibility[] {appVisibility}
    );
    AsyncResult[] arsProfileEmp =
    metadataConnection.create(new Metadata[] {employee});

    Profile manager = new Profile();
    manager.setFullName("Manager");
    manager.setApplicationVisibilities(
        new ProfileApplicationVisibility[] {appVisibility}
    );
    AsyncResult[] arsProfileMgr =
        metadataConnection.create(new Metadata[] {manager});

    // But employees and managers have different access
    // to the state of the expense sheet
    RecordType edit = new RecordType();
    edit.setFullName("ExpenseReport__c.Edit");
    RecordTypePicklistValue editStatuses =
        new RecordTypePicklistValue();
    editStatuses.setPicklist("ExpenseStatus__c");
    editStatuses.setValues(new PicklistValue[]
        {unsubmitted, submitted});
    edit.setPicklistValues(new RecordTypePicklistValue[]
        {editStatuses});
    AsyncResult[] arsRecTypeEdit =
        metadataConnection.create(new Metadata[] {edit});

    RecordType approve = new RecordType();
    approve.setFullName("ExpenseReport__c.Approve");
    RecordTypePicklistValue approveStatuses =
        new RecordTypePicklistValue();
    approveStatuses.setPicklist("ExpenseStatus__c");
    approveStatuses.setValues(new PicklistValue[]
        {approved, rejected});
    approve.setPicklistValues(new RecordTypePicklistValue[]
        {approveStatuses});
    AsyncResult[] arsRecTypeApp =
        metadataConnection.create(new Metadata[] {approve});
} catch (ConnectionException ce) {
```

```
    ce.printStackTrace();
  }
}
```

Declarative Metadata Sample Definition

The definition of a profile in an organization with a custom app, custom object, record type, tab, and user permission is:

```
<?xml version="1.0" encoding="UTF-8"?>
<Profile xmlns="http://soap.sforce.com/2006/04/metadata">
    <applicationVisibilities>
        <application>PubApps__Myriad_Publishing</application>
        <default>false</default>
        <visible>true</visible>
    </applicationVisibilities>
    <custom>true</custom>
    <objectPermissions>
        <object>TestWeblinks__c</object>
        <allowCreate>true</allowCreate>
        <allowDelete>true</allowDelete>
        <allowEdit>true</allowEdit>
        <allowRead>true</allowRead>
        <viewAllRecords>false</viewAllRecords>
        <modifyAllRecords>false</modifyAllRecords>
        <viewAllFields>false</viewAllFields>
    </objectPermissions>
    <recordTypeVisibilities>
        <default>true</default>
        <recordType>TestWeblinks__c.My First Recordtype</recordType>
        <visible>true</visible>
    </recordTypeVisibilities>
    <tabVisibilities>
        <tab>Myriad Publications</tab>
        <visibility>DefaultOn</visibility>
    </tabVisibilities>
    <userPermissions>
        <enabled>true</enabled>
        <name>APIEnabled</name>
    </userpermissions>
</Profile>
```

Usage

To create custom profiles, we recommend that you use the Profile object instead of the `deploy()` call on the Profile Metadata type. The Profile object allows you to create empty profiles that start without any permissions enabled except for required permissions for the profile's user license.

When you use the `retrieve()` call to get information about profiles, the returned `.profile` files only include security settings for the other metadata types referenced in the retrieve request. Exceptions include user permissions, IP address ranges, and login hours, which are always retrieved. For example, the following `package.xml` file contains a `types` element that matches all custom

objects. The returned profiles contain object and field permissions for all custom objects in your organization but don't include permissions for standard objects, such as Account, and standard fields.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>CustomObject</name>
    </types>
   <types>
        <members>*</members>
        <name>Profile</name>
    </types>
    <version>64.0</version>
</Package>
```

The wildcard "*" on CustomObject doesn't match standard objects. This wildcard behavior helps you to avoid making unintended, high-impact profile changes. If you create a few custom objects in a Developer Edition organization, `retrieve()` the information, and later `deploy()` the custom objects to your production org, the profile and field-level security for all your standard objects and fields aren't overwritten. You can only overwrite these standard objects and fields by explicitly creating separate `types` elements for the objects or fields.

Metadata API intentionally makes it difficult to include standard fields in `retrieve()` calls to prevent unexpected profile changes. But you can still retrieve and deploy profile permissions for custom and standard fields in standard objects, such as Account.

This `package.xml` file allows you to return profile permissions for Account standard and custom fields. Note how the standard Account object is defined in a `types` element by specifying it as a member of a CustomObject type.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>Account</members>
        <name>CustomObject</name>
    </types>
   <types>
        <members>*</members>
        <name>Profile</name>
    </types>
    <version>64.0</version>
</Package>
```

This `package.xml` file allows you to return profile permissions for the `MyCustomField__c` custom field in the Account object.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
   <types>
        <members>Account.MyCustomField__c</members>
        <name>CustomField</name>
   </types>
   <types>
        <members>*</members>
        <name>Profile</name>
   </types>
   <version>64.0</version>
</Package>
```

To retrieve field permissions for relationship fields, remove the "Id" part of the field. For example, in this `package.xml` file, to retrieve field permissions for the `AccountId` field for Contacts, you reference this field as `Contact.Account` not `Contact.AcccountId`.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
   <types>
        <members>Contact.Account</members>
        <name>CustomField</name>
   </types>
   <types>
        <members>*</members>
        <name>Profile</name>
   </types>
   <version>64.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SEE ALSO:

Salesforce DX Developer Guide: Retrieve Changes to Profiles with Source Tracking

ProfileActionOverride

Represents an override of an ActionOverride by a user profile. You can use it to override an ActionOverride on a standard Home tab or object record page in Lightning Experience. When a user logs in with a profile, a matching ProfileActionOverride assignment takes precedence over existing overrides for the Home tab or record page specified in ActionOverride. In API versions 39.0 to 44.0, you can access ProfileActionOverride by accessing its encompassing CustomApplication on page 623 or Profile on page 1616 metadata types. In API version 45.0 and later, you can access ProfileActionOverride only by accessing its encompassing CustomApplication on page 623.

Note: ProfileActionOverrides aren't supported in packaging. They're supported in change sets, but you have to add them manually.

File Suffix and Directory Location

Profile-based action overrides are defined as part of a custom application or profile.

Version

ProfileActionOverrides are available in API version 39.0 and later.

ProfileActionOverride can be defined on Profile or CustomApplication for API version 39.0 to 44.0. In API version 45.0 and later, ProfileActionOverride must be defined for CustomApplication instead. Beginning with API version 45.0, Home page assignments related to user profile must also have a corresponding app assignment because more granular Home page assignments are supported. As a result, ProfileActionOverride is defined for CustomApplication rather than Profile.

Fields

Field Name	Field Type	Description
`actionName`	string	The name of the action. The only valid values are `Tab` and `View`. If `pageOrSobjectType` is `standard-home`, this field must be `Tab`. The `Tab` action is supported only when ProfileActionOverride is being specified as part of a Profile in API version 39.0 to 44.0. In API version 45.0 and later, this action is supported only when ProfileActionOverride is being specified as part of a CustomApplication, `pageOrSobjectType` is `standard-home`, and this field is `Tab`. If `pageOrSobjectType` is `record-home`, this field must be `View`. The `View` action is supported only when ProfileActionOverride is being specified as part of a CustomApplication.
`content`	string	Read-only. Represents the name of the Lightning page being used as the override.
`formFactor`	FormFactor (enumeration of type string)	The size of the page being overridden. The `Large` value represents the Lightning Experience desktop environment.
`pageOrSobjectType`	string	The name of the page being overridden. The only valid values are `record-home` and `standard-home`. If the `actionName` is `Tab`, this field must be `standard-home`
`recordType`	string	The record type associated with the override. If `pageOrSobjectType` is `standard-home`, this field must be `null`. This field is required when `actionName` is set to `View`.
`type`	ActionOverrideType (enumeration of type string)	Read-only. The type of action override. The only valid value is `flexipage`.

Usage

You can't delete custom app ProfileActionOverrides by deploying with `destructiveChange.xml`. To delete a ProfileActionOverride, retrieve the app. In the app definition file, find the `<profileActionOverrides>` section, and remove the `<content>` row. Then, change the `<type>` value in that same section to `default` instead of `flexipage`. Do this for every override you want to reset. After making the changes, rezip the folder and deploy.

You can remove one override at a time each with its own deploy, or you can remove multiple overrides in a single deploy. However, we recommend that you do a fresh retrieve every time you want to delete a new override. Don't use a previously retrieved file.

Avoid creating duplicate ProfileActionOverrides in your org. Duplicate ProfileActionOverrides can cause problems, including being unable to select or deselect the **Disable end user personalization of nav items in this app** option in app settings and the **Disable Navigation Bar Personalization in Lightning Experience** User Interface setting.

Declarative Metadata Sample Definition

You can define a ProfileActionOverride like this.

```
<CustomApplication xmlns="http://soap.sforce.com/2006/04/metadata">
    <profileActionOverrides>
        <actionName>View</actionName>
        <content>CustomObjectFlexiPage</content>
        <formFactor>Large</formFactor>
        <pageOrSobjectType>TestObj__c</pageOrSobjectType>
        <type>Flexipage</type>
        <profile>standard</profile>
        <recordType>TestObj__c.TestRecordType</recordType>
    </profileActionOverrides>
    <defaultLandingTab>standard-home</defaultLandingTab>
    <formFactors>Large</formFactors>
    <label>My Custom App</label>
    <tab>standard-Account</tab>
    <tab>standard-Opportunity</tab>
    <uiType>Lightning</uiType>
    <navType>Standard</navType>
</CustomApplication>
```

Here's an example `package.xml`.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>MyCustomApp</members>
        <name>CustomApplication</name>
    </types>
    <version>39.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type doesn't support the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ProfilePasswordPolicy

Represents a profile's password policies. Profile password policies override org-wide password policies for that profile's users. Use ProfilePasswordPolicy to retrieve password policies for a given profile. This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

ProfilePasswordPolicy components have the suffix `.profilePasswordPolicy` and are stored in the `profilePasswordPolicies` folder.

Version

ProfilePasswordPolicy components are available in API version 40.0 and later.

Fields

Field Name	Field Type	Description
`forgotPasswordRedirect`	boolean	If `true`, reset password links in forgot password emails don't immediately expire the first time they're clicked. Instead, the links stay active until a user confirms the password reset request on an interstitial page. The default value is `false`. This field is available in API version 43.0 and later.
`lockoutInterval`	int	Required. The duration of the login lockout, in minutes. If users are locked out, they must wait until the lockout period expires. Valid values: `0`, `15`, `30`, `60`.
`maxLoginAttempts`	int	Required. The number of times a user can enter a wrong password before getting locked out. Valid values: `0`, `3`, `5`, `10`.
`minimumPasswordLength`	int	Required. Minimum number of characters required for a password. Valid values: `5`–`50`.
`minimumPasswordLifetime`	boolean	If `true`, a user cannot change a password more than once in a 24-hour period.
`obscure`	boolean	If `true`, answers to security questions are hidden as the user types.
`passwordComplexity`	int	Required. Level of complexity required for the character types in a user's password. • If `0`, the password can contain any type of character. • If `1`, the password must contain at least one alphabetic character and 1 number. • If `2`, the password must contain at least one alphabetic character, one number, and one of the following special characters: ! # $ % - _ = + < >. • If `3`, the password must contain at least one number, one uppercase letter, and one lowercase letter. • If `4`, the password must contain at least one number, one uppercase letter, one lowercase letter, and one of the following special characters: ! # $ % - _ = + < >.
`passwordExpiration`	int	Required. Number of days until user passwords expire and must be changed. Valid values: • `0`—If set to `0`, the password never expires. • `30` • `60`

Field Name	Field Type	Description
		• 90 • 180 • 365
passwordHistory	int	Required. Number of previous passwords to save. Saving passwords is required to ensure that users reset their password to a new, unique password. This value must be set before a password reset succeeds. If 0, passwordExpiration must be set to 0.
passwordQuestion	int	Required. If set to 1, the answer to the password hint cannot contain the password itself. If 0, the answer has no restrictions.
profile	string	Required. Name of the user profile.

Declarative Metadata Sample Definition

The following is an example of a ProfilePasswordPolicy component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ProfilePasswordPolicy xmlns="http://soap.sforce.com/2006/04/metadata">
    <forgotPasswordRedirect>true</forgotPasswordRedirect>
    <lockoutInterval>30</lockoutInterval>
    <maxLoginAttempts>0</maxLoginAttempts>
    <minimumPasswordLength>7</minimumPasswordLength>
    <minimumPasswordLifetime>false</minimumPasswordLifetime>
    <obscure>false</obscure>
    <passwordComplexity>1</passwordComplexity>
    <passwordExpiration>0</passwordExpiration>
    <passwordHistory>0</passwordHistory>
    <passwordQuestion>1</passwordQuestion>
    <profile>platformportal</profile>
</ProfilePasswordPolicy>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character * (asterisk) in the package.xml manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ProfileSessionSetting

Represents a profile's session settings. Use ProfileSessionSetting to retrieve the session settings for a given profile. This type extends the Metadata metadata type and inherits its fullName field.

File Suffix and Directory Location

ProfileSessionSetting components have the suffix .profileSessionSetting and are stored in the profileSessionSettings folder.

Version

ProfileSessionSetting components are available in API version 40.0 and later.

Fields

Field Name	Field Type	Description
`profile`	string	Required. Name of the user profile.
`requiredSessionLevel`	SessionSecurityLevel	Session security level.
`sessionPersistence`	boolean	Beta. If `true`, keep users logged in to their Experience Cloud site until the session times out—even if they close their browser. Use `sessionPersistence` to reduce how often users must log in to their site. Applies only to the External Identity profile.
`sessionTimeout`	int	Required. Specifies how many minutes of inactivity elapse before a user's authenticated session times out. At the end of the session, the user must log in again. This session timeout value applies to users of the profile and overrides the org-wide timeout value. Changes to the org-wide timeout value don't apply to users of this profile. Valid values: `15`, `30`, `60`, `120`, `240`, `480`, `720`, `1440`.

SessionSecurityLevel

Session security levels control access to certain types of resources based on the type of authentication used for logging in to the current session. For example, username and password authentication requires the `standard` session security level. Multi-factor authentication (MFA) requires `HIGH_ASSURANCE`.

Field Name	Field Type	Description
`SessionSecurityLevel`	(enumeration of type string)	User's security level for the current session. • The `HIGH_ASSURANCE` security level for this session meets the High Assurance requirements set in the org's session settings under Session Security Levels. • The `STANDARD` security level for this session meets the Standard requirements set in the org's session settings under Session Security Levels. • The `LOW` level isn't available or used in the Salesforce UI. It's used at the API level, but users assigned to this level experience unpredictable and reduced functionality.

Declarative Metadata Sample Definition

The following is an example of a ProfileSessionSetting component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ProfileSessionSetting xmlns="http://soap.sforce.com/2006/04/metadata">
```

```
    <profile>platformportal</profile>
    <requiredSessionLevel>HIGH_ASSURANCE</requiredSessionLevel>
    <sessionTimeout>1440</sessionTimeout>
</ProfileSessionSetting>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

Prompt

Represents the metadata related to in-app guidance, which includes prompts and walkthroughs. Help users discover your products and services, adopt your processes, or learn how to use a new feature. Write the content, select the target audience, and specify where and when the in-app guidance appears.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

Prompt components have the suffix `prompt` and are stored in the `prompts` folder.

Version

Prompt components are available in API version 46.0 and later.

Special Access Rules

To add, edit, manage, and view prompts and walkthroughs in Lightning Experience or in Experience Cloud sites, see Considerations for Creating In-App Guidance and Permissions for Creating and Accessing In-App Guidance in *Salesforce Help* for permissions.

Prompts and Walkthroughs in Managed Packages

For considerations about including in-app guidance in a managed package, see Guidelines for In-App Guidance in Managed Packages in Salesforce Help.

For more information about creating managed packages, see Create a First-Generation Managed Package.

Unmanaged packages must contain a namespace prefix. For more information, see Register a Namespace for a First-Generation Managed Packages and What happens to my namespace prefix when I install a package?.

Fields

Field Name	Description
`masterLabel`	**Field Type** string **Description** Required. The label. Maximum of 80 characters.
`promptVersions`	**Field Type** PromptVersion[] **Description** A list of in-app guidance entries. Each entry represents a different prompt or walkthrough.

PromptVersion

A list of in-app guidance entries. Each entry represents a different prompt or walkthrough.

Field Name	Description
`actionButtonLabel`	**Field Type** string **Description** Label for the action button or link. Maximum of 25 characters. For a walkthrough, specify this value on the last step.
`actionButtonLink`	**Field Type** string **Description** URL for the action button or link. Maximum of 1,000 characters. You can't use the `GROUP BY` option in a SOQL query for this field. For a walkthrough, specify this value on the last step.
`body`	**Field Type** string **Description** Required. Body content. In API version 60.0 and later, enter up to 4,000 characters for all prompt types. In earlier API versions, enter up to 240 characters for floating prompts and targeted prompts. Enter up to 4,000 characters for docked prompts. For docked prompts, the maximum characters include HTML markup, not just readable text.

Field Name	Description
`customApplication`	**Field Type** string **Description** Internal use only. No data is populated for this field.
`delayDays`	**Field Type** int **Description** Required if recurrences are scheduled. Number of days in between occurrences. For a walkthrough, specify this value on the first step.
`description`	**Field Type** string **Description** Description. Maximum of 255 characters.
`dismissButtonLabel`	**Field Type** string **Description** Label for the dismiss button of a floating or targeted prompt. Maximum of 15 characters.
`displayPosition`	**Field Type** PromptDisplayPosition (enumeration of type string) **Description** The position of a floating prompt on the page. Valid values are: • `BottomCenter` • `BottomLeft` • `BottomRight` • `TopCenter` • `TopLeft` • `TopRight`
`displayType`	**Field Type** PromptDisplayType (enumeration of type string) **Description** Required. The type of prompt. Valid values are: • `DockedComposer`—A docked prompt • `FloatingPanel`—A floating prompt • `Targeted`—A targeted prompt. Available in API version 52.0 and later.

Field Name	Description
`elementRelativePosition`	**Field Type** PromptElementRelativePosition (enumeration of type string) **Description** Indicates the location of a targeted prompt relative to the element. Available in API version 52.0 and later. Valid values are: • `BottomCenter` • `BottomLeft` • `BottomRight` • `LeftBottom` • `LeftCenter` • `LeftTop` • `RightBottom` • `RightCenter` • `RightTop` • `TopCenter` • `TopLeft` • `TopRight`
`endDate`	**Field Type** date **Description** The date to stop showing the in-app guidance. For a walkthrough, specify this value on the first step.
`header`	**Field Type** string **Description** Label for the header of a docked prompt. This value is the label contained in the window's browser bar. Maximum of 36 characters.
`image`	**Field Type** string **Description** The developer name of the contentAsset that holds the image. You can specify this field or the `imageLink` field, but not both.
`imageAltText`	**Field Type** string **Description** Indicates the alt text of an image. Required if `imageLocation`, `imageLink`, or `image` is specified.

Field Name	Description
`imageLink`	**Field Type** string **Description** The URL for a prompt's image. You can specify this field or the image field, but not both. Available in API version 53.0 and later.
`imageLocation`	**Field Type** picklist **Description** Indicates the location of the image in relation to the body text. Required if `image`, `imageLink`, or `imageAltText` is specified. Valid values are: • `Top` • `Bottom` • `Right`, which is for floating or targeted prompts only • `Left`, which is for floating or targeted prompts only
`indexWithIsPublished`	**Field Type** string **Description** Used by Salesforce for efficient querying.
`indexWithoutIsPublished`	**Field Type** string **Description** Used by Salesforce for efficient querying.
`isPublished`	**Field Type** boolean **Description** Indicates whether the in-app guidance is active (`true`) or not (`false`).
`masterLabel`	**Field Type** string **Description** Required. The label.
`publishedByUser`	**Field Type** string **Description** Internal use only. No data is populated for this field.

Field Name	Description
`publishedDate`	**Field Type** date **Description** Indicates the date the in-app guidance was activated. If installed from a package, this value is the date when the package was installed. For walkthroughs, this field can only be specified on the first step.
`referenceElementContext`	**Field Type** textarea **Description** Used by Salesforce to identify the element that the targeted prompt is associated with. Available in API version 52.0 and later.
`shouldDisplayActionButton`	**Field Type** boolean **Description** Indicates whether an action button or link is included (`true`) or not (`false`).
`shouldIgnoreGlobalDelay`	**Field Type** boolean **Description** Indicates whether the in-app guidance ignores the global time delay and instead shows on page load (`true`) or not (`false`). This field is available in API version 48.0 and later.
`startDate`	**Field Type** date **Description** Indicates the date to start showing the in-app guidance. For a walkthrough, specify this value on the first step. In API version 48.0 and earlier, this field is required.
`stepNumber`	**Field Type** int **Description** Required for walkthroughs only. Indicates the number of the last step the user viewed or interacted with in a walkthrough. Include up to 10 steps. Numbers must be consecutive without repeated or skipped numbers. Available in API version 49.0 and later.
`targetAppDeveloperName`	**Field Type** string **Description** The app's developer name where the in-app guidance appears. Deprecated in API version 51.0 and later.

Field Name	Description
`targetAppNamespacePrefix`	**Field Type** string **Description** The app's namespace prefix where the in-app guidance appears. This value must match the target app's `NamespacePrefix` in the org that the package is being installed into. Maximum of 15 characters. Deprecated in API version 51.0 and later.
`targetPageKey1`	**Field Type** string **Description** Required. Used by Salesforce to identity the prompt's page location along with `targetPageKey2`, `targetPageKey3`, `targetPageKey4`, and `targetPageType`.
`targetPageKey2`	**Field Type** string **Description** Used by Salesforce to identity the prompt's page location along with `targetPageKey1`, `targetPageKey3`, `targetPageKey4`, and `targetPageType`.
`targetPageKey3`	**Field Type** string **Description** Used by Salesforce to identify the prompt's page location along with `targetPageKey1`, `targetPageKey2`, `targetPageKey4`, and `targetPageType`.
`targetPageKey4`	**Field Type** string **Description** Used by Salesforce to identify the prompt's page location along with `targetPageKey1`, `targetPageKey2`, `targetPageKey3`, and `targetPageType`. This field is available in API version 53.0 and later.
`targetPageType`	**Field Type** string **Description** Required. Used by Salesforce to identity the page location along with `targetPageKey1`, `targetPageKey2`, `targetPageKey3`, and `targetPageKey4`.
`targetRecordType`	**Field Type** string

Field Name	Description
	Description Used by Salesforce to determine if in-app guidance is specific to a record type. This field is available in API version 53.0 and later.
`themeColor`	**Field Type** PromptThemeColor (enumeration of type string) **Description** Indicates which custom theme color is applied to the in-app guidance. Required if `themeSaturation` is specified. For a walkthrough, specify this value on the first step. Valid values are: • `Theme1`, which is derived from the current brand color • `Theme2`, which is derived from the current page background color • `Theme3`, which is derived from the current global header color • `Theme4`, which is derived from the current app theme color
`themeSaturation`	**Field Type** PromptThemeSaturation (enumeration of type string) **Description** Indicates which color value, or saturation, is applied to the in-app guidance that has a custom theme color. Required if `themeColor` is specified. For a walkthrough, specify this value on the first step. Valid values are: • `Dark` • `Light`
`timesToDisplay`	**Field Type** int **Description** Required if recurrences are scheduled. The maximum number of times to show the in-app guidance. Salesforce detects whether the user interacts with the in-app guidance, then determines whether to show the in-app guidance again or cancel scheduled recurrences. Maximum value of 30. For a walkthrough, specify this value on the first step.
`title`	**Field Type** string **Description** Required. The label for the title. Maximum of 36 characters.
`uiFormulaRule`	**Field Type** UiFormulaRule[] **Description** A set of one or more permission filters that define the conditions under which the in-app guidance displays on the page.

Field Name	Description
	If the rule evaluates to `true`, the in-app guidance displays on the page. If `false`, it doesn't display. If this field is `null`, the in-app guidance displays by default.
`userAccess`	**Field Type** PromptUserAccess (enumeration of type string) **Description** Indicates which permissions can see the in-app guidance. Valid values are: • `Everyone`, which indicates that there's no permission restrictions • `SpecificPermissions`, which indicates that only users with all the specific user permissions specified can see the in-app guidance In API version 48.0 and earlier, this field is required.
`userProfileAccess`	**Field Type** PromptUserProfileAccess (enumeration of type string) **Description** Indicates which profiles can see the in-app guidance. This field is available in API version 48.0 and later. Valid values are: • `Everyone`, which indicates that there are no profile restrictions • `SpecificProfiles`, which indicates that users with any of the specified user profiles can see the in-app guidance
`versionNumber`	**Field Type** int **Description** Required. The number remains `1` since multiple versions aren't saved in the org.
`videoLink`	**Field Type** string **Description** The embed URL for a video in a docked prompt. Maximum of 1,000 characters. You can specify this field or the `image` field, but not both. This field is available in API version 48.0 and later. See Considerations for Creating In-App Guidance.

UiFormulaRule

A set of one or more filters that define the conditions under which a prompt displays on a Lightning Experience page.

Field Name	Description
`booleanFilter`	**Field Type** string

Field Name	Description
	Description Specifies the AND filter condition.
`criteria`	**Field Type** UiFormulaCriterion[] **Description** List of one or more filters that, when evaluated, determine visibility.

UiFormulaCriterion

A single filter that, when evaluated, helps define visibility on a Lightning Experience page.

Field Name	Description
`leftValue`	**Field Type** string **Description** Required. The field used for filtering. Only standard and custom permissions can be included. You can use these expressions in the `leftValue` field when setting filters for visibility. • `{!$Permission.CustomPermission.`***`permissionName`***`}`—Use this expression to control visibility based on the custom permissions of the user viewing the Lightning page. Supported for app, Home, and record pages only. • `{!$Permission.StandardPermission.`***`permissionName`***`}`—Use this expression to control visibility based on the standard permissions of the user viewing the Lightning page. Supported for app, Home, and record pages only. • `{!ENCODED:{!ID:$`***`User.Profile.Key`***`}}`—Use this expression to control visibility based on the custom or standard profile of the user viewing the Lightning page. Available in API Version 48.0 and later.
`operator`	**Field Type** string **Description** Required. Defines the operator used to filter the data. Valid value is `EQUAL`.
`rightValue`	**Field Type** string

Field Name	Description
	Description Specifies if you want to evaluate the visibility for permissions or the name of the profile. • For permissions, use `true`. • For profiles, use the name of the profile. Available in API Version 48.0 and later. For example, `Standard` or `custom_regionalsales`.

Declarative Metadata Sample Definition

The following is an example of a Prompt component.

```
<?xml version="1.0" encoding="UTF-8"?>
<Prompt xmlns="http://soap.sforce.com/2006/04/metadata">
    <masterLabel>Prompt Label</masterLabel>
    <promptVersions>
        <actionButtonLabel>Learn How</actionButtonLabel>

<actionButtonLink>https://trailhead.salesforce.com/en/content/learn/modules/scrum-and-kanban-at-salesforce/learn-about-kanban</actionButtonLink>

        <body>Explore how the Path and the Kanban view can help you track, manage, and
update your records.</body>
        <delayDays>1</delayDays>
        <description>Kanban floating prompt</description>
        <dismissButtonLabel>OK</dismissButtonLabel>
        <displayPosition>TopLeft</displayPosition>
        <displayType>FloatingPanel</displayType>
        <endDate>2019-03-11</endDate>
        <isPublished>true</isPublished>
        <masterLabel>Prompt Label</masterLabel>
        <publishedDate>2019-03-11</publishedDate>
        <shouldDisplayActionButton>false</shouldDisplayActionButton>
        <shouldIgnoreGlobalDelay>false</shouldIgnoreGlobalDelay>
        <startDate>2019-03-11</startDate>
        <targetAppDeveloperName>LightningSales</targetAppDeveloperName>
        <targetAppNamespacePrefix>standard</targetAppNamespacePrefix>
        <timesToDisplay>3</timesToDisplay>
        <title>Get on the Path to Success</title>
        <userAccess>SpecificPermissions</userAccess>
        <userProfileAccess>SpecificProfiles</userProfileAccess>
        <versionNumber>1</versionNumber>
        <videolink>https://www.youtube.com/embed/Ko-gcObzTVo</videolink>
        <uiFormulaRule>
            <booleanFilter>(1 AND 2 AND 3) AND (4 OR 5)</booleanFilter>
            <criteria>
                <leftValue>{!$Permission.StandardPermission.ActivitiesAccess}</leftValue>

                <operator>EQUAL</operator>
                <rightValue>TRUE</rightValue>
            </criteria>
```

```
            <criteria>
               <leftValue>{!$Permission.StandardPermission.ContentWorkspaces}</leftValue>

                <operator>EQUAL</operator>
                <rightValue>TRUE</rightValue>
            </criteria>
            <criteria>
                <leftValue>{!$Permission.CustomPermission.MyCustomPerm}</leftValue>
                <operator>EQUAL</operator>
                <rightValue>TRUE</rightValue>
            </criteria>
            <criteria>
                <leftValue>{!ENCODED:{!ID:$User.Profile.Key}}</leftValue>
                <operator>EQUAL</operator>
                <rightValue>Standard</rightValue>
            </criteria>
            <criteria>
                <leftValue>{!ENCODED:{!ID:$User.Profile.Key}}</leftValue>
                <operator>EQUAL</operator>
                <rightValue>custom_mysysadmin</rightValue>
            </criteria>
        </uiFormulaRule>
    </promptVersions>
</Prompt>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>Prompt</name>
    </types>
    <version>46.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

PublicKeyCertificate

Represents the public key certificate. On this entity we store a public certificate or a JSON web key, which is used to validate the customer-provided JWT.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

PublicKeyCertificate components have the suffix `.PublicKeyCertificate` and are stored in the `PublicKeyCertificate` folder.

Version

PublicKeyCertificate components are available in API version 62 and later.

Special Access Rules

There are no additional access requirements that are specific to this type.

Fields

Field Name	Description
`description`	**Field Type** string **Description** A description of the public key certificate.
`isActive`	**Field Type** boolean **Description** Indicates whether the public key certificate is active (true) or inactive (false). The default value is false.
`jsonWebKey`	**Field Type** string **Description** Represents a public cryptographic key that can be used to verify the validity of a token.
`masterLabel`	**Field Type** string **Description** Required. The label for the public key certificate.

Declarative Metadata Sample Definition

The following is an example of a PublicKeyCertificate component.

```
<?xml version="1.0" encoding="UTF-8"?>
<PublicKeyCertificate xmlns="http://soap.sforce.com/2006/04/metadata">
```

```
    <masterLabel>pck1</masterLabel>
    <isActive>true</isActive>
    <description>This is my description for a test PublicKeyCertificate</description>
    <jsonWebKey>
{
  "kid":"123456",
  "alg":"RS256",
  "use":"sig",
  "kty":"RSA",
  "x5c":["<Your public certificate>"],
  "y":"y",
  "n":"<Base64-encoded modulus>",
  "e":"<Base64-encoded public exponent>",
  "crv":"crv",
  "d":"d",
  "k":"k"
}
    </jsonWebKey>
</PublicKeyCertificate>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
  ~ Copyright 2024 salesforce.com, inc.
  ~ All Rights Reserved
  ~ Company Confidential
  -->

<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>PublicKeyCertificate</name>
    </types>
    <version>62.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

PublicKeyCertificateSet

Represents a set of public certificate keys. On this entity we store a public certificates or JSON web keys.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

PublicKeyCertificateSet components have the suffix `.PublicKeyCertificateSet` and are stored in the `PublicKeyCertificateSet` folder.

Version

PublicKeyCertificateSet components are available in API version 62 and later.

Special Access Rules

There are no additional access requirements that are specific to this type.

Fields

Field Name	Description
`description`	**Field Type** string **Description** A description of the public key certificate set.
`jwksEndPoint`	**Field Type** string **Description** The URL of the HTTPS Server that returns the JWKS.
`jwtIssuer`	**Field Type** string **Description** The user, organization, or service that issued the JSON web token. This value is case-sensitive.
`masterLabel`	**Field Type** string **Description** Required. The label for the public key certificate set.
`publicKeyCertificateSetKeys`	**Field Type** PublicKeyCertificateSetKey[] **Description** A set of public certificate keys associated with the public key certificate set.

Field Name	Description
type	**Field Type** PublicKeyCertificateSetType (enumeration of type string) **Description** Required. Determines how the server's public key set is retrieved. The keys are represented in JWK format. Values are: • JWKS- Used to specify a certificate via the child Type PublicKeyCertificateSetKey. • JWKS_URL- Used to specify a certificate via the jwksEndPoint field on this Type.

PublicKeyCertificateSetKeys

Represents a set of public certificate keys associated with the public key certificate set.

Field Name	Description
publicKeyCertificate	**Field Type** string **Description** Required. The PublicKeyCertificate we want to reference.

Declarative Metadata Sample Definition

The following is an example of a PublicKeyCertificateSet component.

```
<?xml version="1.0" encoding="UTF-8"?>
<PublicKeyCertificateSet xmlns="http://soap.sforce.com/2006/04/metadata">
    <masterLabel>pcks1</masterLabel>
    <description>This is my description for a PublicKeyCertificateSet</description>
    <type>JWKS</type>
    <jwtIssuer>example.com</jwtIssuer>
    <publicKeyCertificateSetKeys>
            <publicKeyCertificate>pck1</publicKeyCertificate>
    </publicKeyCertificateSetKeys>
</PublicKeyCertificateSet>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>PublicKeyCertificate</name>
    </types>
```

```
    <types>
        <members>*</members>
        <name>PublicKeyCertificateSet</name>
    </types>
    <version>62.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

Queue

Represents a holding area for items before they are processed.

Declarative Metadata File Suffix and Directory Location

The file suffix for queue components is `.queue` and components are stored in the `queues` directory of the corresponding package directory. This component supports cases, leads, service contracts (if Entitlements are enabled), and custom objects.

Version

Queue components are available in API version 24.0 and later.

Special Access Rules

As of Summer '20 and later, only authenticated internal and external users can access this type.

Fields

This metadata type represents the valid values that define a queue:

Field Name	Field Type	Description
`doesSendEmailToMembers`	boolean	Indicates whether emails are sent to queue members (`true`) or not (`false`) when a new record is added to the queue.
`email`	string	The email address of the queue owner.
`name`	string	Required. The name of the queue. Corresponds to **Label** in the user interface.
`queueMembers`	QueueMembers[]	Represents queue members added to the queue. Members can be added directly or selected by roles and public groups they belong to. Available in API version 42.0 and later.
`queueRoutingConfig`	string	Routing configuration name. Applies to orgs that use Omni-Channel with a routing configuration. Available in API version 42.0 and later.

Field Name	Field Type	Description
queueSobject	QueueSobject[]	Indicates the supported entity types.

QueueMembers

Represents queue members added to the queue. Members can be added directly as users or selected by the roles and public groups they belong to. Available in API version 42.0 and later.

Field Name	Field Type	Description
publicGroups	PublicGroups[]	Represents public groups in the org. Public groups are optionally used to add queue members.
roleAndSubordinates	RoleAndSubordinates[]	Represents roles and their subordinates in the org's role hierarchy, including customer and partner roles. Roles and their subordinate hierarchy are optionally used to add queue members.
roleAndSubordinatesInternal	RoleAndSubordinatesInternal[]	Represents internal roles and their subordinates in the org's role hierarchy, excluding customer and partner roles. Roles and their subordinate hierarchy are optionally used to add queue members.
roles	Roles[]	Represents roles in the org. Roles are optionally used to add queue members.
users	Users[]	Represents users in the org. Users can be added directly as queue members.

PublicGroups

Represents public groups in the org. Public groups are optionally used to add queue members. Available in API version 42.0 and later.

Field Name	Field Type	Description
publicGroup	string	Represents a public group.

RoleAndSubordinates

Represents roles and their subordinates in the org's role hierarchy, including customer and partner roles. Roles and their subordinate hierarchy can be used to add queue members. Available in API version 42.0 and later.

Field Name	Field Type	Description
roleAndSubordinate	string	Represents a role and its subordinates, including customer and partner roles. The availability of this field depends on the type of org, org creation date, release update enforcement status, and whether digital experiences is enabled. For more information, see this knowledge article.

RoleAndSubordinatesInternal

Represents internal roles and their subordinates in the org's role hierarchy, excluding customer and partner roles. Roles and their subordinate hierarchy can be used to add queue members. Available in API version 42.0 and later.

Field Name	Field Type	Description
`roleAndSubordinateInternal`	string	Represents a role and its subordinates, excluding customer and partner roles. The availability of this field depends on the type of org, org creation date, release update enforcement status, and whether digital experiences is enabled. For more information, see this knowledge article.

Roles

Represents roles in the org. Roles can be used to add queue members. Available in API version 42.0 and later.

Field Name	Field Type	Description
`role`	string	Represents a role.

Users

Represents users in the org. Users can be added directly as queue members. Available in API version 42.0 and later.

Field Name	Field Type	Description
`user`	string	Represents a user. Specify the user's username.

QueueSobject

QueueSobject represents an entity type that the queue supports.

Field Name	Field Type	Description
`sobjectType`	string	Valid values are: • `Case` • `ContactRequest` • `Lead` • `ServiceContract` • `Task` (Available in API version 48.0 and later.) • Custom objects (such as `ObjA__c`)

Declarative Metadata Sample Definition

The following is the definition of a queue, which supports Case, Lead, and a custom object named ObjA.

```
<?xml version="1.0" encoding="UTF-8"?>
<Queue xmlns="http://soap.sforce.com/2006/04/metadata">
    <doesSendEmailToMembers>true</doesSendEmailToMembers>
    <email>member@company.com</email>
    <fullName>Your Name</fullName>
    <name>memberQueue</name>
    <queueSobject>
        <sobjectType>Case</sobjectType>
    </queueSobject>
    <queueSobject>
        <sobjectType>Lead</sobjectType>
    </queueSobject>
    <queueSobject>
        <sobjectType>ObjA__c</sobjectType>
    </queueSobject>
</Queue>
```

Here's another definition of a queue containing queue members added directly or via public groups and roles. Queries retrieve values using the `DeveloperName` field, not the `Name` field, so that the returned names are unique. The query also appends letters to the end of duplicate names, so these groups and roles can be referred to independently.

```
<?xml version="1.0" encoding="UTF-8"?>
<Queue xmlns="http://soap.sforce.com/2006/04/metadata">
    <doesSendEmailToMembers>false</doesSendEmailToMembers>
    <name>queue1</name>
    <queueMembers>
        <publicGroups>
            <publicGroup>All Internal Users</publicGroup>
        <publicGroups>
        <queueRoleAndSubordinates>
            <queueRoleAndSubordinate>role1</queueRoleAndSubordinate>
            <queueRoleAndSubordinate>role2</queueRoleAndSubordinate>
            <queueRoleAndSubordinate>role3</queueRoleAndSubordinate>
        </queueRoleAndSubordinates>
        <roles>
            <role>role1</role>
        </roles>
        <users>
            <user>s@sm.com</user>
            <user>std@sm.com</user>
        </users>
    </queueMembers>
    <queueRoutingConfig>my_omni_routing_config</queueRoutingConfig>
    <queueSobject>
        <sobjectType>Case</sobjectType>
    </queueSobject>
    <queueSobject>
        <sobjectType>Lead</sobjectType>
    </queueSobject>
</Queue>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

QueueRoutingConfig

Represents the settings that determine how work items are routed to agents.

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

ServicePresenceStatus components have the suffix `.queueRoutingConfig` and are stored in the `queueRoutingConfigs` folder.

Version

QueueRoutingConfig components are available in API version 44.0 and later.

Special Access Rules

This type is available only if Omni-Channel is enabled in your org.

Fields

Field Name	Field Type	Description
`capacityPercentage`	double	The percentage of an agent's capacity for work items that's consumed by a specific type of work item from this service channel. For example, you might give phone calls a capacity percentage of 100. If an agent receives a phone call, the agent won't receive new work items until the call ends, because at that point the agent's capacity will have reached 100%.
capacityType	CapacityType	The setting applies for PSRs (PendingServiceRouting) that are created and managed by the system. • When set to `INHERITED`, the value of the Interruptible check box or value set on the Service Channel applies. • When set to `INTERRUPTIBLE`, the generated PSR has the `isInterruptible` flag set to true. • When set to NOT `INTERRUPTIBLE`, the generated PSR has the `isInterruptible` flag set to false. • When not set, its behavior is equivalent to `INHERITED`. -

Field Name	Field Type	Description
`capacityWeight`	double	The amount of an agent's capacity for work items that's consumed by a work item from this service channel. For example, if an agent has a capacity of 6, and cases are assigned a capacity weight of 2, an agent can be assigned up to 3 cases before the agent is at capacity and can't receive new work items.
`dropAdditionalSkillsTimeout`	int	The number of seconds to elapse before additional skills are dropped from Omni-Channel routing. In skills-based routing, you can set some skills to **Additional Skill**. After the timeout elapses, a skill marked as **Additional Skill** is dropped from Omni-Channel routing and the case is routed to the best-matched agent, even if the agent doesn't have all the skills. If CustomRequestedDateTime is set in the PendingServiceRouting object, DropAdditionalSkillsTimeout uses CustomRequestedDateTime as the start time. If CustomRequestedDateTime + DropAdditionalSkillsTimeout has already passed, Omni-Channel immediately drops the additional skills after the pending service request is created.
`isAttributeBased`	boolean	Indicates whether this routing configuration is used with skills-based routing rules (`true`) or not (`false`).
`label`	string	Required. The label of the presence status.
`PausedCapacityPercentage`	double	The percentage of a rep's capacity that's consumed when this work item is paused. The paused capacity feature is available with status-based capacity and Enhanced Omni-Channel only. Available in API version 64.0 and later.
`PausedCapacityWeight`	double	The amount of a rep's capacity that's consumed when this work item is paused. The paused capacity feature is available with status-based capacity and Enhanced Omni-Channel only. Available in API version 64.0 and later.
`pushTimeout`	int	The number of seconds set for push timeout. `0` is returned when push timeout isn't enabled.
`queueOverflowAssignee`	string	The ID of the queue that's set as the Overflow Assignee.
`QueueRoutingConfigSkill`	QueueRoutingConfigSkill[]	Default skills associated with the routing configuration. Work is routed using a combination of rules and default skills.
`routingModel`	RoutingModel (enumeration of type string)	Required. The routing type that determines how work items are routed (pushed) to agents. Possible values are: • `LEAST_ACTIVE` • `MOST_AVAILABLE` • `EXTERNAL_ROUTING`
`routingPriority`	int	Required. The priority in which work items from the service channels that are related to this routing configuration are routed to agents. Work

Field Name	Field Type	Description
		items from routing configurations that have lower priority values (for example, 0) are routed to agents first.
`userOverflowAssignee`	string	The ID of the user that's set as the Overflow Assignee.

QueueRoutingConfigSkill

Represents default skills associated with the routing configuration.

Fields

Field Name	Field Type	Description
`skill`	string	Skill used to route a work item.

Declarative Metadata Sample Definition

The following is an example of a QueueRoutingConfig component.

```
<?xml version="1.0" encoding="UTF-8"?>
<QueueRoutingConfig xmlns="http://soap.sforce.com/2006/04/metadata">
  <capacityWeight>1.0</capacityWeight>
  <label>Case Routing</label>
  <pushTimeout>120</pushTimeout>
  <queueOverflowAssignee>queueOverflow</queueOverflowAssignee>
  <routingModel>LEAST_ACTIVE</routingModel>
  <routingPriority>1</routingPriority>
  <capacityType>INHERITED</capacityType>
  <pausedCapacityWeight>0.25</pausedCapacityWeight>
</QueueRoutingConfig>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>QueueRoutingConfig</name>
    </types>
    <version>44.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

QuickAction

Represents a specified create or update quick action for an object that then becomes available in the Chatter publisher. For example, you can create an action that, on the detail page of an account, allows a user to create a contact related to that account from the Chatter feed on that page. QuickAction can be created on objects that permit custom fields.

The parent objects supported include:

- Account
- Campaign
- Case
- Contact
- ContentNote
- Custom objects
- Group
- Lead
- Opportunity

File Suffix and Directory Location

QuickAction components have the suffix `quickAction` and are stored in the `quickActions` folder.

Version

QuickAction components are available in API version 28.0 and later.

Fields

Field Name	Field Type	Description
`canvas`	string	If the custom action invokes a Canvas app, the app name. Returns the fully qualified name of the Canvas app in the format `<namespace>__<dev_name>`, if the quick action type is `Canvas`; otherwise, returns `null`. This field is available in API version 29.0 and later.
`description`	string	The description of the action.
`fieldOverrides`	FieldOverride on page 1665[]	The specific field that can be overridden within a QuickAction on page 1662.
`flowDefinition`	string	If the custom action invokes a flow, this field represents the API name of the flow. Otherwise, this field is `null`.
`height`	int	If a custom action is created, this field represents the height in pixels of the action pane.

Field Name	Field Type	Description
`icon`	string	The icon used to identify the action. API version 32.0 and later returns different icons than in earlier API versions.
`isProtected`	boolean	Indicates whether this component is protected (`true`) or not (`false`). Protected components cannot be linked to or referenced by components created in the installing organization.
`label`	string	Identifies the action and displays to users. Also the default identifier used for the API and managed packages.
`lightningComponent`	string	If the custom action invokes a Lightning component, this field represents the fully qualified name of the component. Otherwise, this field is `null`. Available in API version 38.0 and later.
`optionsCreateFeedItem`	boolean	Required. Indicates whether successful completion of the action creates a feed item (`true`) or not (`false`). Applies only to Create Record, Update Record, and Log a Call quick action types. Available in API version 36.0 and later.
`page`	string	If the custom action invokes a Visualforce page, this field identifies the page.
`quickActionLayout`	QuickActionLayout	The layout of fields on the action.
`quickActionParameter`	QuickActionParameter[]	The input and output of the quick action. Available in API version 63.0 and later.
`standardLabel`	QuickActionLabel (enumeration of type string)	Specifies the standard label to use for the action. The valid values are: • `AddRecord` • `AddMember` • `ChangeDueDate` • `ChangePriority` • `ChangeStatus` • `CreateNew` • `CreateNewRecordType` (For example, a label with something like “Create New Idea”) • `Defer` • `EditDescription` • `EnrollInProgram` (Available in API versions 46.0 and later only if the org has Health Cloud enabled) • `Escalate` • `EscalateToRecord` • `Forward` (Available in API version 42.0 and later)

Field Name	Field Type	Description
		• `LogACall` • `LogANote` • `ModifyAppointment` (Available in API version 47.0 and later) • `New` (A new record) • `NewChild` (A new child record) • `NewChildRecordType` • `NewRecordType` (For example, a label with something like "New Idea") • `OfferFeedback` • `PatientDetails` (Available in API version 57.0 and later if the org has Health Cloud enabled) • `Quick` (A quick record) • `QuickRecordType` • `Reply` (Available in API version 42.0 and later) • `ReplyAll` (Available in API version 42.0 and later) • `RequestFeedback` • `SendEmail` (This value is available in API version 31.0 and later.) • `Update`
`successMessage`	string	The message that displays to the user upon successful completion of the action. Available in API version 36.0 and later.
`targetObject`	string	The object for which the action is created and performed. For example, you can create an action that, on the detail page of an account, allows a user to create a contact related to that account from the Chatter feed on that page. QuickAction can be created on objects that permit custom fields. In this case, Contact is the `targetObject`.
`targetParentField`	string	The parent object type of the action. Links the target object to the parent object. For example, use Account if the target object is Contact and the parent object is Account.
`targetRecordType`	string	Specifies which record type to create. Valid values are: • Business Account • Person Account • Master
`type`	QuickActionType (enumeration of type string)	Required. The type of quick action. Valid values are: • `Canvas` • `Create` • `Flow` (This value is available as a Beta in API version 41.0 and later)

Field Name	Field Type	Description
		• `LightningComponent` (This value is available in API version 38.0 and later.) • `LogACall` • `Post` • `SendEmail` (This value is available in API version 31.0 and later.) • `SocialPost` • `Update` • `VisualforcePage`
`width`	int	If a custom action is created, this field represents the width in pixels of the action pane.

FieldOverride

Represents the field names and their respective formulas and literal values that comprise predefined value settings for a QuickAction on page 1662. If a field on an action has both a predefined value and a default value set, the action uses the predefined value, not the default value. A formula value takes precedence over a literal value if both are defined.

Field Name	Field Type	Description
`field`	string	Required. The name of the field to allow predefined values on.
`formula`	string	Specifies the formula to use when setting a field's predefined value. Supported for single-select picklists as of API version 43.0.
`literalValue`	string	Supported for picklists only. Specifies the literal value of the field defined from values in the picklist. Corresponds to the Specific Value field in the predefined value UI.

QuickActionLayout

The layout of fields on the action. There's no hard limit to the number of fields you can add to an action layout. However, for optimum usability, we recommend a maximum of eight fields. Adding more than 20 fields can severely affect user efficiency.

Field Name	Field Type	Description
`layoutSectionStyle`	LayoutSectionStyle (enumeration of type string)	Required. The type of layout structure used. The valid values are: • `TwoColumnsTopToBottom` • `TwoColumnsLeftToRight` • `OneColumn` • `CustomLinks`
`quickActionLayoutColumns`	QuickActionLayoutColumn on page 1666[]	Specifies columns in a QuickActionLayout on page 1665.

QuickActionLayoutColumn

A column defined for a QuickActionLayout on page 1665.

Field Name	Field Type	Description
`quickActionLayoutItems`	QuickActionLayoutItem on page 1666 []	Specifies row items in a QuickActionLayoutColumn on page 1666.

QuickActionLayoutItem

A row item comprised of fields and defined for a QuickActionLayoutColumn on page 1666.

Field Name	Field Type	Description
`emptySpace`	boolean	Controls if this layout item is a blank space (`true`) or not (`false`).
`field`	string	Represents a specific field in QuickActionLayoutItem on page 1666.
`uiBehavior`	UiBehavior (enumeration of type string)	Specifies user input behavior for specific fields in QuickActionLayoutItem on page 1666. The valid values are: • `Edit` • `Required` • `Readonly`

QuickActionParameter

Represents the input and output of the associated quick action. Available in API version 63.0 and later.

Field Name	Field Type	Description
`name`	string	Required. Name of the parameter.
`type`	QuickActionParameterType (enumeration of type string)	Required. `Input` is the only valid value.
`value`	string	Represents the value associated with the given parameter name.

Declarative Metadata Sample Definition

The following is an example of a QuickAction on page 1662 component:

```
<?xml version="1.0" encoding="UTF-8"?>
<QuickAction xmlns="http://soap.sforce.com/2006/04/metadata">
    <description>testActionDefinitionTypesCreateTask</description>
    <label>testActionDefinitionTypesCreateTask</label>
    <optionsCreateFeedItem>true</optionsCreateFeedItem>
```

```
    <quickActionLayout>
        <layoutSectionStyle>TwoColumnsLeftToRight</layoutSectionStyle>
        <quickActionLayoutColumns>
            <quickActionLayoutItems>
                <emptySpace>false</emptySpace>
                <field>OwnerId</field>
                <uiBehavior>Required</uiBehavior>
            </quickActionLayoutItems>
            <quickActionLayoutItems>
                <emptySpace>false</emptySpace>
                <field>WhoId</field>
                <uiBehavior>Edit</uiBehavior>
            </quickActionLayoutItems>
            <quickActionLayoutItems>
                <emptySpace>false</emptySpace>
                <field>WhatId</field>
                <uiBehavior>Edit</uiBehavior>
            </quickActionLayoutItems>
            <quickActionLayoutItems>
                <emptySpace>false</emptySpace>
                <field>ActivityDate</field>
                <uiBehavior>Edit</uiBehavior>
            </quickActionLayoutItems>
            <quickActionLayoutItems>
                <emptySpace>false</emptySpace>
                <field>Subject</field>
                <uiBehavior>Edit</uiBehavior>
            </quickActionLayoutItems>
            <quickActionLayoutItems>
                <emptySpace>false</emptySpace>
                <field>Status</field>
                <uiBehavior>Required</uiBehavior>
            </quickActionLayoutItems>
            <quickActionLayoutItems>
                <emptySpace>false</emptySpace>
                <field>Priority</field>
                <uiBehavior>Required</uiBehavior>
            </quickActionLayoutItems>
        </quickActionLayoutColumns>
        <quickActionLayoutColumns/>
    </quickActionLayout>
    <successMessage>This is a success message</successMessage>
    <targetObject>Task</targetObject>
    <targetParentField>What</targetParentField>
    <type>Create</type>
</QuickAction>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

RedirectWhitelistUrl

Represents a trusted URL that's excluded from redirection restrictions when the `redirectionWarning` or `redirectBlockModeEnabled` field on the SessionSettings Metadata type is set to `true`. This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. Because changing terms in our code can break current implementations, we maintained this metadata type's name.

File Suffix and Directory Location

RedirectWhitelistUrl components have the suffix `.redirectWhitelistUrl` and are stored in the `redirectWhitelistUrls` folder.

Version

RedirectWhitelistUrl components are available in API version 48.0 and later.

Special Access Rules

Only authenticated internal and external users with the View Setup and Configuration permission can access this type, and only users with the Customize Application permission can edit it.

Fields

Field Name	Field Type	Description
`url`	string	Required. The trusted URL. These formats are accepted: `example.com`, `*.example.com`, and `https://example.com`. The host section of the URL can include an asterisk (`*`) as a wildcard. Otherwise, the URL cannot be malformed. Examples of malformed URLs that fail a syntax check are `malformed^url.example.com`, and `https://{subdomain}.example.com`. To add a `URL` based on parameters, build the URL before you add it to this Metadata Type.

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

RecommendationStrategy

Represents a recommendation strategy. Recommendation strategies are applications, similar to data flows, that determine a set of recommendations to be delivered to the client through data retrieval, branching, and logic operations.

File Suffix and Directory Location

RecommendationStrategy components have the suffix `.recommendationStrategy` and are stored in the `recommendationStrategies` folder.

Version

RecommendationStrategy components are available in API version 45.0 and later.

Special Access Rules

Metadata access for the RecommendationStrategy type is backed by the ManageRecommendationStrategies user permission.

Fields

Field Name	Field Type	Description
`actionContext`	StrategyAction[]	An array of action contexts used by the strategy.
`contextRecordType`	string	The sObject type of the $record used by the flow.
`description`	string	Description of the recommendation strategy.
`filter`	StrategyNodeFilter[]	An array of filter nodes.
`if`	StrategyNodeIf[]	An array of if nodes.
`invocableAction`	StrategyNodeInvocableAction[]	An array of Apex invocable action nodes. Available in API version 46.0 and later.
`isTemplate`	boolean	Indicates whether the recommendation strategy is a template (`true`) or not (`false`). When installed from managed packages, recommendation strategies can't be viewed or cloned by subscribers because of intellectual property (IP) protection. But when those recommendation strategies are templates, subscribers can open them in a builder, clone them, and customize the clones. The default value of this field is `false`. Available in API version 47.0 and later.
`label`	string	Required. Label for the flow.
`map`	StrategyNodeMap[]	An array of map nodes. Available in API version 46.0 and later.
`mutuallyExclusive`	StrategyNodeExclusive[]	An array of mutuallyExclusive nodes.

Field Name	Field Type	Description
onBehalfOfExpression	string	Formula expression defining the intended target of the recommendations (in other words, the Contact associated with a Case). Mainly used for reaction tracking.
recommendationLimit	StrategyNodeRecommendationLimit[]	An array of recommendation limit nodes.
recommendationLoad	StrategyNodeRecommendationLoad[]	An array of recommendation load nodes.
sort	StrategyNodeSort[]	An array of sort nodes.
union	StrategyNodeUnion[]	An array of union nodes.

StrategyNodeBase

Base class for all strategy nodes. This is an abstract class.

Field Name	Field Type	Description
childNode	string	Array of child node names, in order of execution.
description	string	Description of the node.
label	string	Label of the node.
name	string	Required. Unique name of the node.

StrategyAction

Defines a call to an invocable action from the strategy. Results are used by decision elements in the strategy.

Field Name	Field Type	Description
action	string	Required. The name or id of the InvocableAction to execute.
argument	StrategyActionArg[]	List of strategy action arguments.
description	string	Description of the strategy.
label	string	Label for the strategy action.
name	string	Required. Unique name of the strategy action, which is referenced by decisioning elements in the strategy.
type	InvocableActionType (enumeration of type string)	Required. The action type. Valid values are: • activateSessionPermSet—Activates a session-based permission set for the running user. • addMessageToChat—Adds a message to an existing Salesforce Anywhere chat. This value is available in API version 49.0 and later. • addMessageToQuipChat—Adds a Quip message to an existing chat room. This value is available in API version 46.0 and later.

Field Name	Field Type	Description
		• `addMessageToQuipDocument`—Adds a Quip message to an existing Quip document, spreadsheet, or slide. This value is available in API version 46.0 and later. • `addQuipDocumentToFolder`—Adds an existing Quip document, spreadsheet, or slide to an existing folder. This value is available in API version 46.0 and later. • `addUsersToChat`—Adds users to an existing Salesforce Anywhere chat. This value is available in API version 49.0 and later. • `addUsersToQuipDocument`—Adds users, identified by their email addresses, to an existing Quip document, spreadsheet, or slide. This value is available in API version 46.0 and later. • `addUsersToQuipChat`—Adds users, identified by their email addresses, to an existing Quip chat room. This value is available in API version 46.0 and later. • `answerQuestionsWithSalesforceDocumentation`—Searches Salesforce documentation to answer questions and provide links to relevant articles. • `attachQuipDocumentToRecord`—Attaches a Quip document, spreadsheet, or slide to a Salesforce record. This value is available in API version 46.0 and later. • `apex`—Invokes an Apex method that has the @invocableMethod annotation. • `archiveKnowledgeArticles`—Archives a list of published Knowledge articles. This value is available in API version 45.0 and later. • `assignKnowledgeArticles`—Mass assigns knowledge articles from article list views. This value is available in API version 44.0 and later. • `cdpRunIdentityResolution`—Runs a Data Cloud identity resolution process. This value is available in API version 57.0 and later. • `chat`—Creates a Salesforce Anywhere chat. This value is available in API version 49.0 and later. • `chatterPost`—Posts to Chatter. • `choosePricebook`—Selects a price book. • `contactRequestAction`—Creates a contact request record. This value is available in API version 45.0 and later. • `component`—Invokes the Lightning component that implements the `lightning:availableForFlowActions` interface and that is referenced by `actionName`. This value is available in API version 43.0 and later. • `contentWorkspaceEnableFolders`—Enables folders in a library. • `copyQuipDocument`—Creates a copy of an existing Quip document, spreadsheet, or slide, and gives it a new title. This value is available in API version 46.0 and later.

Field Name	Field Type	Description
		• `createDraftFromOnlineKnowledgeArticle`—Creates a draft from a published knowledge article. This value is available in API version 45.0 and later. • `createInvoiceFromFulfillmentOrder`—Creates an invoice from a purchase order. Available to B2B Commerce. This value is available in API version 49.0 and later. • `createQuipChat`—Creates a Quip chat room. This value is available in API version 46.0 and later. • `createQuipDocument`—Creates a Quip document, spreadsheet, or slide. This value is available in API version 46.0 and later. • `createQuipFolder`—Creates a Quip folder. This value is available in API version 46.0 and later. • `customNotificationAction`—Sends a custom notification. This value is available in API version 46.0 and later. • `deactivateSessionPermSet`—Deactivates a session-based permission set for the running user. • `deleteKnowledgeArticle`—Deletes a draft version (translation or master-language) or an entire archived knowledge article. This value is available in API version 46.0 and later. • `dynamicSendSurveyInvitation`—Sends customized notifications to users about important events or updates to the records that they're working on. This value is available in API version 51.0 and later. • `editQuipDocument`—Modifies the contents of an existing Quip document, spreadsheet, or slide. This value is available in API version 46.0 and later. • `emailAlert`—Sends an email by referencing a workflow email alert • `emailSimple`—Sends an email by using flow resources • `exploreConversation`—Retrieves insights from a conversation. This value is available in API version 61.0 and later. • `externalConnector`—Executes a process or method exposed via a connector to an external system. This value is available in API version 63.0 and later. • `externalService`—Invokes an External Service operation that makes an HTTP request to an external system made available by an External Service schema registered through Setup. This value is available in API version 46.0 and later. • `findMatchingIndividuals`—Finds contact, lead, or employee records that match a search term. • `flow`—Invokes an autolaunched flow. This action type isn't available for flows with a processType of Flow or AutolaunchedFlow. To invoke an autolaunched flow from one of those types, use FlowSubflow. This value is available in API version 32.0 and later.

Field Name	Field Type	Description
		• `generateAiAgentResponse`—Generates a response from the AI agent based on input and instructions to support intelligent, conversational experiences. This value is available in API version 63.0 and later. • `generateVerificationCode`—Sends a verification code to the customer's email to verify their identity. This value is available in API version 63.0 and later. • `getArticleSmartLinkUrl`—Gets the Smart Link URL of the Salesforce Knowledge article. Smart links go to the right article and version, even when a new version is published or the URL name changes. This value is available in API version 54.0 and later. • `goToCadenceStep`—Jumps to the specified step in the Sales cadence. This value is available in API version 57.0 and later. • `internalTestAction`—Reserved for internal use. • `internalTestConnectApiAction`—Reserved for internal use. • `limitRepetitions`—Limit the number of times the same recommendation or offer appears on the same record or for the same user during a time period in a recommendation strategy flow. This value is available in API version 55.0 and later. • `massUpdateAccountForecast`—Bulk updates forecasts asynchronously. This value is available in API version 48.0 and later. • `massUpdateSalesAgreement`—Bulk updates sales agreements asynchronously. This value is available in API version 48.0 and later. • `quickAction`—Invokes a QuickAction. • `publishActionableOrchSrcEvent`—Publishes events triggered by an external system. This value is available in API version 62.0 and later. • `publishKnowledgeArticles`—Mass publishes knowledge articles from article list views. This value is available in API version 44.0 and later. • `restoreKnowledgeArticleVersion`—Restores an archived version of a knowledge article. This value is available in API version 45.0 and later. • `sendAlert`—Sends Salesforce Anywhere alerts to users. This value is available in API version 49.0 and later. • `sendNotification`—Sends an available notification type. This value is available in API version 54.0 and later. • `sendSurveyInvitation`—Sends email survey invitations to leads, contacts, and users in your org based on an action, such as when a customer support case closes. This value is available in API version 47.0 and later. • `performSurveySentimentAnalysis`—Perform survey sentiment analysis to create or update the AI Sentiment Result records. This value is available in API version 55.0 and later. • `skillsBasedRouting`—Creates a PendingServiceRouting record used for Omni-Channel skills-based routing. This value is available in version 44.0 and later.

Field Name	Field Type	Description
		• `slackArchiveChannel`—Archives a Slack channel in a Slack workspace. This value is available in API version 54.0 and later. • `slackCheckUsersAreConnectedToSlack`—Indicates whether a collection of Salesforce users is connected to a given Slack app. This value is available in API version 54.0 and later. • `slackCreateChannel`—Creates a Slack channel in a Slack workspace. This value is available in API version 54.0 and later. • `slackGetConversationInfo`—Retrieves the name of a Slack channel or group direct message and finds out whether it's archived. This value is available in API version 54.0 and later. • `slackInviteUsersToChannel`—Adds users who are connected to a given Slack app to a Slack channel or group direct message. This value is available in API version 54.0 and later. • `slackPinMessage`—Pin or unpin a message in a Slack channel or group direct message. This value is available in API version 54.0 and later. • `slackPostMessage`—Send a message to a Slack channel or group direct message. This value is available in API version 54.0 and later. • `slackSendMessageToLaunchFlow`—Send a message to a Slack channel, direct message, or the Messages tab of a Slack app that includes a button that a recipient can use to launch a screen flow. This value is available in API version 55.0 and later. • `slackUpdateMessage`—Edits a message that was previously sent to a Slack channel or group direct message. This value is available in API version 54.0 and later. • `submitKnowledgeArticleForTranslation`—Submits a published or draft knowledge article for translation. This value is available in API version 46.0 and later. • `submit`—Submits a record for approval. • `verifyCustomerCode`—Verifies the code entered by the customer to complete identity verification. This value is available in API version 49.0 and later. These values are used in Omnichannel Inventory. If no version is specified, the value is available in API version 51.0 and later. • `ociCreateReservation`—Creates one or more inventory reservations at a location or location group. • `ociFulfillReservation`—Fulfills one or more inventory reservations at a location. • `ociGetAvailability`—Gets inventory availability data for one or more products at one or more inventory locations or location groups. • `ociReleaseReservation`—Releases one or more inventory reservations. • `ociTransferReservation`—Transfers one or more inventory reservations between locations or location groups.

Field Name	Field Type	Description
		These values are used in the B2B Commerce Checkout Flow. If no version is specified, the value is available in API version 47.0 and later. • `updateCheckoutSessionStateAction`—Updates the checkout session next state for checkout flows. This value is available in API version 49.0 and later. • `priceCart`—Requests prices for all items in a cart during B2B Commerce checkout. This value is available in API version 47.0 and later. • `checkoutSessionAction`—Initiates or retrieves an existing Checkout Session for Checkout Flows. Available to B2B Commerce. This value is available in API version 49.0 and later. • `cancelCartAsyncOperation`—Cancels a WebCart's async operation. Available to B2B Commerce. This value is available in API version 49.0 and later. • `calcCartPromotionsAction`—Requests a full cart promotion calculation of all applicable line items in the Web Cart during B2B Commerce checkout. This value is available in API version 52.0 and later. • `checkCartInventoryAction`—Requests an inventory for all items in a Web Cart during B2B Commerce checkout. This value is available in API version 47.0 and later. • `calcCartShipmentAction`—Calculates the shipping cost for all items in a Web Cart during B2B Commerce checkout. This value is available in API version 47.0 and later. • `cartToOrderAction`—Creates a Salesforce Standard Order in draft mode. This value is available in API version 47.0 and later. • `activateOrderAction`—Activates a draft order, which creates an order summary. This value is available in API version 47.0 and later. For values used in Business Rules Engine, see Flow for Business Rules Engine. These values are used in the Commerce Checkout Flow. If no version is specified, the value is available in API version 55.0 and later. • `addCartItem`—Adds an item to a cart during Commerce checkout. • `createCart`—Creates a cart during Commerce checkout. • `deleteCart`—Deletes a cart during Commerce checkout. These values are used in Salesforce CMS Workflows and Approvals. If no version is specified, the value is available in API version 58.0 and later. • `managedContentPublishVariant`—Publishes a content variant associated with a flow. This value is available in API version 59.0 and later. • `managedContentRoleStepInteractive`—Assigns a content variant review to a CMS role. • `managedContentUnpublishVariant`—Unpublishes a published content variant associated with a flow. This value is available in API version 59.0 and later.

Field Name	Field Type	Description
		• `managedContentVariantSetLockStatus`—Sets the locked status of a content variant. • `managedContentVariantSetReadyStatus`—Sets the ready for publication status of a content variant. These values are used in Employee Service. If no version is specified, the value is available in API version 63.0 and later. • `createServiceRequestCase`—Creates a case or incident for the requested service. • `getLeaveBalance`—Gets the leave balance of a specific employee. These values are used in Insurance. If no version is specified, the value is available in API version 63.0 and later. • `cancelInsurancePolicy`—Cancel an insurance policy by using a set of user inputs that represent policy details. • `endorseInsurancePolicy`—Endorse an insurance policy by using a set of user inputs that represent policy details. • `getInsurancePolicy`—Get the details of an insurance policy. • `issueInsurancePolicy`— Issue an insurance policy by using a set of user inputs that represent policy details. • `renewInsurancePolicy`—Renew an insurance policy by using a set of user inputs that represent policy details. These values are used in Insurance Brokerage. If no version is specified, the value is available in API version 63.0 and later. • `computeProducerSplits`—Compute the producer splits for the producers associated with an Insurance Policy, for a Commission Statement Line Item. • `createProducerCommissions`—Create records for the commissions that producers receive for the insurance policy associated with the specified commission statement line item, and update the commission statement line item record status. • `findInsurancePolicy`—Get the insurance policy associated with a commission statement line item that matches the specified criteria, and update the status of the commission statement line item record. These values are used in Order Management. If no version is specified, the value is available in API version 48.0 and later. • `addOrderItemSummarySubmit`—Adds order item summaries to an order summary. This value is available in API version 54.0 and later. • `adjustOrderItemSummariesPreview`—Previews the expected results of applying a price adjustment to order item summaries from an order summary without actually applying it. This value is available in API version 49.0 and later.

Field Name	Field Type	Description
		• `adjustOrderItemSummariesSubmit`—Applies a price adjustment to order item summaries from an order summary. This value is available in API version 49.0 and later. • `authorizePayment`—Authorizes a card payment. This value is available in API version 55.0 and later. • `cancelFulfillmentOrderItem`—Removes items from a fulfillment order. • `cancelOrderItemSummariesPreview`—Previews the expected results of canceling order item summaries from an order summary without actually canceling them. • `cancelOrderItemSummariesSubmit`—Cancels order item summaries from an order summary. • `confirmHeldFulfillmentOrderCapacity`—Confirms held fulfillment order capacity. This value is available in API version 55.0 and later. • `createCreditMemoOrderSummary`—Creates a credit memo for an order summary. • `createFieldGnrnPromptTmplResp`—Creates a field generation prompt template response. This value is available in API version 62.0 and later. • `createFulfillmentOrder`—Creates one or more fulfillment orders and fulfillment order products for an order delivery group summary, which defines a recipient and delivery method. • `createFulfillmentOrders`—Creates fulfillment orders and fulfillment order products for multiple order delivery group summaries, each of which defines a recipient and delivery method. This value is available in API version 51.0 and later. • `createInvoiceFromChangeOrders`—Creates an invoice for one or more change orders. This value is available in API version 56.0 and later. • `createInvoiceFromFulfillmentOrder`—Creates an invoice for a fulfillment order. • `createOrderPaymentSummary`—Creates an order payment summary for an authorization or payments belonging to an order summary. • `createOrderSummary`—Creates an order summary for an order. • `createReturnOrder`—Creates a return order and return order items for an order. • `ensureFundsOrderSummaryAsync`—Triggers an asynchronous background process to ensure funds through a payment provider for an invoice belonging to an order summary. • `ensureRefundsOrderSummaryAsync`—Triggers an asynchronous background process to ensure refunds through a payment provider for an invoice belonging to an order summary.

Field Name	Field Type	Description
		• `getFulfillmentOrderCapacityValues`—Gets fulfillment order capacity information. This value is available in API version 55.0 and later. • `holdFulfillmentOrderCapacity`—Holds fulfillment order capacity. This value is available in API version 55.0 and later. • `orderRoutingFindRoutesWithFewestSplits`—Evaluates ordered product quantities against available inventory to determine the smallest combination of locations that can fulfill the order. This value is available in API version 51.0 and later. • `orderRoutingFindRoutesWithFewestSplitsUsingOCI`—Evaluates ordered product quantities against available inventory at specified location groups and locations to determine the smallest combination of locations that can fulfill the order. This value is available in API version 54.0 and later. • `orderRoutingRankByAverageDistance`—Calculates the average distance from sets of inventory locations to an order recipient, and returns the sets sorted by that average distance. This value is available in API version 51.0 and later. • `releaseHeldFulfillmentOrderCapacity`—Releases held fulfillment order capacity. This value is available in API version 55.0 and later. • `returnOrderItemSummariesPreview`—Previews the expected results of returning order item summaries from an order summary without actually returning them. • `returnOrderItemSummariesSubmit`—Returns order item summaries from an order summary. • `returnReturnOrderItems`—Processes return order line items. For values used in Financial Services Cloud, see Flow for Financial Services Cloud. For values used in Fundraising for Nonprofit Cloud, see Flow for Fundraising. For values used in Health Cloud, see Flow for Health Cloud. For values used in Manufacturing Cloud, see Flow for Manufacturing Cloud. These values are used in Rebate Management. • `addRebateMemberList`—Adds a list of members to a rebate program. This value is available in API version 51.0 and later. • `calculateProjectedRebateAmount`—Calculates the projected rebate amount for rebate types associated with a specified transaction ID. This value is available in API version 54.0 and later. • `calculateRebateAmountAndUpsertPayout`—Calculates the rebate amount and upserts the rebate payout for the specified aggregate record. This value is available in API version 51.0 and later.

Field Name	Field Type	Description
		• `getBenefitAndCalculateRebateAmount`—Gets benefit details, and optionally calculates the rebate amount for the specified aggregate record. This value is available in API version 51.0 and later. • `getEligibleProgramRebateTypes`—Retrieves the eligible program rebate types for a mapped object. This value is available in API version 52.0 and later. • `generateRebatePayoutPeriods`—Generates payout periods for a rebate program based on the frequency specified in the program. This value is available in API version 51.0 and later. • `processRebatesBatchCalculationJob`—Processes a rebate batch calculation job from the Data Processing Engine. This value is available in API version 51.0 and later. • `processProgramRebateTypeProducts`—Insert or delete records in the Program Rebate Type Product object. This value is available in API version 53.0 and later. • `rebatesProcessCSV`—Processes an uploaded CSV file using Bulk API 2.0 and converts the file's data into records in the target object. This value is available in API version 51.0 and later. • `upsertCustomRebatePayout`—Upserts the custom calculated rebate payout for the specified aggregate record. This value is available in API version 51.0 and later. These values are used in Loyalty Management. • `adjustPoints`—Adjusts loyalty points for a specified program member or journal transaction. This value is available in API version 51.0 and later. • `assignTierBenefits`— Assigns Member Benefits to a member tier for benefits that are associated with a Benefit Action. This value is available in API version 51.0 and later. • `cancelAccrual`—Cancels a specific set of accrual transactions. • `creditPoints`—Credits loyalty points to a specified program member's balance. This value is available in API version 51.0 and later. • `cancelRedemption`—Reverts a specific set of redemption transactions. This value is available in API version 51.0 and later. • `changeTier`—Changes the tier for a specified program member. This value is available in API version 51.0 and later. • `changeTierWhenNoErrors`—Changes tier for a specified loyalty program member only when all the input parameters meet the criteria. This value is available in API version 51.0 and later. • `debitPoints`—Debits loyalty points to a specified program member's balance. This value is available in API version 51.0 and later. • `executeMemberBenefit`—Processes the benefit action associated with the benefit, which is assigned to a loyalty program member. This value is available in API version 51.0 and later.

Field Name	Field Type	Description
		• `generateMemberReferralCode`—Generates a unique 8-character referral code for a loyalty program member. This value is available in API version 57.0 and later. • `getMemberActiveSegments`—Retrieve active Data Cloud market segments that a loyalty program member is a part of. • `getTier`—Gets the current tier for a specified program member. This value is available in API version 51.0 and later. • `getPointsBalance`—Gets the loyalty points balance for a specified program member. This value is available in API version 51.0 and later. • `getLoyaltyPromotion`—Gets active loyalty promotions based on a transaction journal. This value is available in API version 53.0 and later. • `getLoyaltyPromotionBasedOnSalesforceCDP`—Gets promotions for a member based on the market segment the member belongs to. This value is available in API version 53.0 and later. • `issueVoucher`—Issues a voucher for a member or contract. This value is available in API version 51.0 and later. • `mergeLoyaltyProgramMembership`—Merges two active loyalty program member records that both belong to the same loyalty program. This value is available in API version 56.0 and later. • `transferMemberPointsToGroups`—Transfers points from an individual member or a corporate member to the member's associated group. This value is available in API version 53.0 and later. • `updateProgressForCumulativePromotionUsage`—Updates the progress a member has made towards attaining a cumulative type promotion. This value is available in API version 53.0 and later. • `unmergeLoyaltyProgramMembership`—Unmerges loyalty program member records that have a Merged status. The action unmerges memberships in the Merged status from the previously merged membership. This value is available in API version 56.0 and later. • `runProgramProcess`—Triggers an active loyalty program process. This value is available in API version 56.0 and later. • `runProgramProcessForTransactionJournal`—Triggers an active loyalty program process whose process type is TransactionJournal. This value is available in API version 54.0 and later. These values are for Decision Table. • `decisionTableAction`—Runs an active decision table definition. This value is available in API version 51.0 and later. • `refreshDecisionTable`—Refreshes the decision table cache. This value is available in API version 51.0 and later. These values are for the Batch Management jobs. • `batchJobAction`—Runs the batch management jobs definitions. This value is available in API version 51.0 and later.

Field Name	Field Type	Description
		• `submitFailedRecordsBatchJob`—Resubmits an existing batch job with failed records for processing. This value is available in API version 52.0 and later. This value is for Data Processing Engine. • `dataProcessingEngineAction`—Runs the data processing engine definitions. This value is available in API version 51.0 and later. This value is used for Einstein Visit Recommendation. • `saveRecommendationDecision`—Save visit and task recommendation decisions. This value is available in API version 51.0 and later. These values are used in Field Service. If no version is specified, the value is available in API version 52.0 and later. • `addWorkPlans`—Creates work plan and work step objects from the work plan library. • `addWorkSteps`—Creates work step objects from the work plan library. • `deleteWorkPlans`—Deletes all the work plans and work steps associated with a work order or work order line item. • `generateWorkPlans`—Generates work plans based off rules defined in the work plan library. For values used in Intelligent Form Reader, see Flow for Intelligent Form Reader. For values used in Intelligent Document Reader, see Flow for Intelligent Document Reader. This value is used in Public Sector Solutions. • `createBenefitDisbursement`—Creates a benefit disbursement for an eligible benefit assignment. This value is available in API version 57.0 and later. • `runRecordAggrBatchProcDef`—Runs a Data Processing Engine definition to process an asynchronous batch job that creates or updates record aggregation results. This value is available in API version 59.0 and later. These values are reserved for future use. • `thanks` • `metricRefresh` • `exportSurveyResponses`

StrategyActionArg

Defines arguments passed to invocable actions associated with a strategy action.

Field Name	Field Type	Description
`name`	string	Required. Unique name for the parameter to pass to the invocable action.
`value`	string	Required. A Salesforce formula expression that is evaluated with the result being used as the parameter value for the Strategy Action.

StrategyNodeUnionBase

Base class for nodes that perform a union of their children. Union nodes combine the outputs of their children to form the input to themselves. StrategyNodeUnionBase extends StrategyNodeBase and inherits all of its fields. This is an abstract class.

Field Name	Field Type	Description
`limit`	int	Maximum number of results to output.

StrategyNodeFilter

Defines a filter element that filters recommendations. It extends StrategyNodeUnionBase and inherits all its fields.

Field Name	Field Type	Description
`expression`	string	Required. A formula expression that results in a boolean value when executed on each recommendation in the node's input. Inputs that result in `true` form the output, and inputs that result in `false` are excluded.

StrategyNodeIf

Selects specific children to execute and combines their results. Executes and returns results of children based on the array of child node expressions. Extends StrategyNodeUnionBase and inherits all of its fields.

Field Name	Field Type	Description
`childNodeExpression`	IfExpression[]	Array of if expressions.
`onlyFirstMatch`	boolean	If `true`, selects only the results from the matching child. If `false`, selects and combines results from all matching children. The default value is `false`.

IfExpression

Expression used by StrategyNodeIf.

Field Name	Field Type	Description
`childName`	string	Required. Name of child to match.
`expression`	string	Required. Formula expression returning `true` or `false`.

StrategyNodeInvocableAction

Defines an element that calls an Apex invocable action to generate or enhance a list of recommendations. It extends StrategyNodeUnionBase and inherits all its fields.

Field Name	Field Type	Description
`action`	string	Required. The name of the invocable action to execute.
`argument`	StrategyNodeInvocableActionArg[]	List of arguments that are passed to the invocable action.
`isGenerator`	boolean	Required. If `true`, the UI displays the Generate element. If `false`, the UI displays the Enhance element. Defaults to `false`.
`type`	InvocableActionType (enumeration of type string)	Required. The action type. Valid values are: • `activateSessionPermSet`—Activates a session-based permission set for the running user. • `addMessageToChat`—Adds a message to an existing Salesforce Anywhere chat. This value is available in API version 49.0 and later. • `addMessageToQuipChat`—Adds a Quip message to an existing chat room. This value is available in API version 46.0 and later. • `addMessageToQuipDocument`—Adds a Quip message to an existing Quip document, spreadsheet, or slide. This value is available in API version 46.0 and later. • `addQuipDocumentToFolder`—Adds an existing Quip document, spreadsheet, or slide to an existing folder. This value is available in API version 46.0 and later. • `addUsersToChat`—Adds users to an existing Salesforce Anywhere chat. This value is available in API version 49.0 and later. • `addUsersToQuipDocument`—Adds users, identified by their email addresses, to an existing Quip document, spreadsheet, or slide. This value is available in API version 46.0 and later. • `addUsersToQuipChat`—Adds users, identified by their email addresses, to an existing Quip chat room. This value is available in API version 46.0 and later. • `answerQuestionsWithSalesforceDocumentation`—Searches Salesforce documentation to provide answer to questions, as well as links to relevant articles. • `attachQuipDocumentToRecord`—Attaches a Quip document, spreadsheet, or slide to a Salesforce record. This value is available in API version 46.0 and later. • `apex`—Invokes an Apex method that has the @invocableMethod annotation. • `archiveKnowledgeArticles`—Archives a list of published Knowledge articles. This value is available in API version 45.0 and later. • `assignKnowledgeArticles`—Mass assigns knowledge articles from article list views. This value is available in API version 44.0 and later.

Field Name	Field Type	Description
		• `cdpRunIdentityResolution`—Runs a Data Cloud identity resolution process. This value is available in API version 57.0 and later. • `chat`—Creates a Salesforce Anywhere chat. This value is available in API version 49.0 and later. • `chatterPost`—Posts to Chatter. • `choosePricebook`—Selects a price book. • `contactRequestAction`—Creates a contact request record. This value is available in API version 45.0 and later. • `component`—Invokes the Lightning component that implements the `lightning:availableForFlowActions` interface and that is referenced by `actionName`. This value is available in API version 43.0 and later. • `contentWorkspaceEnableFolders`—Enables folders in a library. • `copyQuipDocument`—Creates a copy of an existing Quip document, spreadsheet, or slide, and gives it a new title. This value is available in API version 46.0 and later. • `createDraftFromOnlineKnowledgeArticle`—Creates a draft from a published knowledge article. This value is available in API version 45.0 and later. • `createInvoiceFromFulfillmentOrder`—Creates an invoice from a purchase order. Available to B2B Commerce. This value is available in API version 49.0 and later. • `createQuipChat`—Creates a Quip chat room. This value is available in API version 46.0 and later. • `createQuipDocument`—Creates a Quip document, spreadsheet, or slide. This value is available in API version 46.0 and later. • `createQuipFolder`—Creates a Quip folder. This value is available in API version 46.0 and later. • `customNotificationAction`—Sends a custom notification. This value is available in API version 46.0 and later. • `deactivateSessionPermSet`—Deactivates a session-based permission set for the running user. • `deleteKnowledgeArticle`—Deletes a draft version (translation or master-language) or an entire archived knowledge article. This value is available in API version 46.0 and later. • `dynamicSendSurveyInvitation`—Sends customized notifications to users about important events or updates to the records that they're working on. This value is available in API version 51.0 and later. • `editQuipDocument`—Modifies the contents of an existing Quip document, spreadsheet, or slide. This value is available in API version 46.0 and later. • `emailAlert`—Sends an email by referencing a workflow email alert • `emailSimple`—Sends an email by using flow resources

Field Name	Field Type	Description
		• `exploreConversation`—Retrieves insights from a conversation. This value is available in API version 61.0 and later. • `externalService`—Invokes an External Service operation that makes an HTTP request to an external system made available by an External Service schema registered through Setup. This value is available in API version 46.0 and later. • `findMatchingIndividuals`—Finds contact, lead, or employee records that match a search term. • `flow`—Invokes an autolaunched flow. This action type isn't available for flows with a processType of Flow or AutolaunchedFlow. To invoke an autolaunched flow from one of those types, use FlowSubflow. This value is available in API version 32.0 and later. • `generateAiAgentResponse`—Generates a response from the AI agent based on input and instructions to support intelligent, conversational experiences. This value is available in API version 63.0 and later. • `generateVerificationCode`—Sends a verification code to the customer's email to verify their identity. This value is available in API version 63.0 and later. • `getAgentConvTscp`—Retrieves the transcript of conversations between the agent and the customer. This value is available in API version 50.0 and later. • `getArticleSmartLinkUrl`—Gets the Smart Link URL of the Salesforce Knowledge article. Smart links go to the right article and version, even when a new version is published or the URL name changes. This value is available in API version 54.0 and later. • `internalTestAction`—Reserved for internal use. • `internalTestConnectApiAction`—Reserved for internal use. • `limitRepetitions`—Limit the number of times the same recommendation or offer appears on the same record or for the same user during a time period in a recommendation strategy flow. This value is available in API version 55.0 and later. • `massUpdateAccountForecast`—Bulk updates forecasts asynchronously. This value is available in API version 48.0 and later. • `massUpdateSalesAgreement`—Bulk updates sales agreements asynchronously. This value is available in API version 48.0 and later. • `quickAction`—Invokes a QuickAction. • `parseConvoAnalysis`—Parses conversation data to analyze sentiment or extract actionable insights. This value is available in API version 51.0 and later. • `publishActionableOrchSrcEvent`—Publishes events triggered by an external system. This value is available in API version 62.0 and later. • `publishKnowledgeArticles`—Mass publishes knowledge articles from article list views. This value is available in API version 44.0 and later.

Field Name	Field Type	Description
		• `restoreKnowledgeArticleVersion`—Restores an archived version of a knowledge article. This value is available in API version 45.0 and later. • `sendAlert`—Sends Salesforce Anywhere alerts to users. This value is available in API version 49.0 and later. • `sendNotification`—Sends an available notification type. This value is available in API version 54.0 and later. • `sendSurveyInvitation`—Sends email survey invitations to leads, contacts, and users in your org based on an action, such as when a customer support case closes. This value is available in API version 47.0 and later. • `performSurveySentimentAnalysis`—Perform survey sentiment analysis to create or update the AI Sentiment Result records. This value is available in API version 55.0 and later. • `skillsBasedRouting`—Creates a PendingServiceRouting record used for Omni-Channel skills-based routing. This value is available in version 44.0 and later. • `slackArchiveChannel`—Archives a Slack channel in a Slack workspace. This value is available in API version 54.0 and later. • `slackCheckUsersAreConnectedToSlack`—Indicates whether a collection of Salesforce users is connected to a given Slack app. This value is available in API version 54.0 and later. • `slackCreateChannel`—Creates a Slack channel in a Slack workspace. This value is available in API version 54.0 and later. • `slackGetConversationInfo`—Retrieves the name of a Slack channel or group direct message and finds out whether it's archived. This value is available in API version 54.0 and later. • `slackInviteUsersToChannel`—Adds users who are connected to a given Slack app to a Slack channel or group direct message. This value is available in API version 54.0 and later. • `slackPinMessage`—Pin or unpin a message in a Slack channel or group direct message. This value is available in API version 54.0 and later. • `slackPostMessage`—Send a message to a Slack channel or group direct message. This value is available in API version 54.0 and later. • `slackSendMessageToLaunchFlow`—Send a message to a Slack channel, direct message, or the Messages tab of a Slack app that includes a button that a recipient can use to launch a screen flow. This value is available in API version 55.0 and later. • `slackUpdateMessage`—Edits a message that was previously sent to a Slack channel or group direct message. This value is available in API version 54.0 and later. • `submitKnowledgeArticleForTranslation`—Submits a published or draft knowledge article for translation. This value is available in API version 46.0 and later.

Field Name	Field Type	Description
		• `submit`—Submits a record for approval. • `verifyCustomerCode`—Verifies the code entered by the customer to complete identity verification. This value is available in API version 63.0 and later. These values are used in Omnichannel Inventory. If no version is specified, the value is available in API version 51.0 and later. • `ociCreateReservation`—Creates one or more inventory reservations at a location or location group. • `ociFulfillReservation`—Fulfills one or more inventory reservations at a location. • `ociGetAvailability`—Gets inventory availability data for one or more products at one or more inventory locations or location groups. • `ociReleaseReservation`—Releases one or more inventory reservations. • `ociTransferReservation`—Transfers one or more inventory reservations between locations or location groups. These values are used in the B2B Commerce Checkout Flow. If no version is specified, the value is available in API version 47.0 and later. • `updateCheckoutSessionStateAction`—Updates the checkout session next state for checkout flows. This value is available in API version 49.0 and later. • `priceCart`—Requests prices for all items in a cart during B2B Commerce checkout. This value is available in API version 47.0 and later. • `checkoutSessionAction`—Initiates or retrieves an existing Checkout Session for Checkout Flows. Available to B2B Commerce. This value is available in API version 49.0 and later. • `cancelCartAsyncOperation`—Cancels a WebCart's async operation. Available to B2B Commerce. This value is available in API version 49.0 and later. • `calcCartPromotionsAction`—Requests a full cart promotion calculation of all applicable line items in the Web Cart during B2B Commerce checkout. This value is available in API version 52.0 and later. • `checkCartInventoryAction`—Requests an inventory for all items in a Web Cart during B2B Commerce checkout. This value is available in API version 47.0 and later. • `calcCartShipmentAction`—Calculates the shipping cost for all items in a Web Cart during B2B Commerce checkout. This value is available in API version 47.0 and later. • `cartToOrderAction`—Creates a Salesforce Standard Order in draft mode. This value is available in API version 47.0 and later. • `activateOrderAction`—Activates a draft order, which creates an order summary. This value is available in API version 47.0 and later.

Field Name	Field Type	Description
		For values used in Business Rules Engine, see Flow for Business Rules Engine. These values are used in the Commerce Checkout Flow. If no version is specified, the value is available in API version 55.0 and later. • `addCartItem`—Adds an item to a cart during Commerce checkout. • `createCart`—Creates a cart during Commerce checkout. • `deleteCart`—Deletes a cart during Commerce checkout. These values are used in Salesforce CMS Workflows and Approvals. If no version is specified, the value is available in API version 58.0 and later. • `managedContentPublishVariant`—Publishes a content variant associated with a flow. This value is available in API version 59.0 and later. • `managedContentRoleStepInteractive`—Assigns a content variant review to a CMS role. • `managedContentUnpublishVariant`—Unpublishes a published content variant associated with a flow. This value is available in API version 59.0 and later. • `managedContentVariantSetLockStatus`—Sets the locked status of a content variant. • `managedContentVariantSetReadyStatus`—Sets the ready for publication status of a content variant. These values are used in Employee Service. If no version is specified, the value is available in API version 63.0 and later. • `createServiceRequestCase`—Creates a case or incident for the requested service. • `getLeaveBalance`—Gets the leave balance of a specific employee. These values are used in Insurance. If no version is specified, the value is available in API version 63.0 and later. • `cancelInsurancePolicy`—Cancel an insurance policy by using a set of user inputs that represent policy details. • `endorseInsurancePolicy`—Endorse an insurance policy by using a set of user inputs that represent policy details. • `getInsurancePolicy`—Get the details of an insurance policy. • `issueInsurancePolicy`— Issue an insurance policy by using a set of user inputs that represent policy details. • `renewInsurancePolicy`—Renew an insurance policy by using a set of user inputs that represent policy details. These values are used in Insurance Brokerage. If no version is specified, the value is available in API version 63.0 and later. • `computeProducerSplits`—Compute the producer splits for the producers associated with an Insurance Policy, for a Commission Statement Line Item.

Field Name	Field Type	Description
		• `createProducerCommissions`—Create records for the commissions that producers receive for the insurance policy associated with the specified commission statement line item, and update the commission statement line item record status. • `findInsurancePolicy`—Get the insurance policy associated with a commission statement line item that matches the specified criteria, and update the status of the commission statement line item record. These values are used in Order Management. If no version is specified, the value is available in API version 48.0 and later. • `addOrderItemSummarySubmit`—Adds order item summaries to an order summary. This value is available in API version 54.0 and later. • `adjustOrderItemSummariesPreview`—Previews the expected results of applying a price adjustment to order item summaries from an order summary without actually applying it. This value is available in API version 49.0 and later. • `adjustOrderItemSummariesSubmit`—Applies a price adjustment to order item summaries from an order summary. This value is available in API version 49.0 and later. • `authorizePayment`—Authorizes a card payment. This value is available in API version 55.0 and later. • `cancelFulfillmentOrderItem`—Removes items from a fulfillment order. • `cancelOrderItemSummariesPreview`—Previews the expected results of canceling order item summaries from an order summary without actually canceling them. • `cancelOrderItemSummariesSubmit`—Cancels order item summaries from an order summary. • `confirmHeldFulfillmentOrderCapacity`—Confirms held fulfillment order capacity. This value is available in API version 55.0 and later. • `createCreditMemoOrderSummary`—Creates a credit memo for an order summary. • `createFieldGnrnPromptTmplResp`—Creates a field generation prompt template response. This value is available in API version 62.0 and later. • `createFulfillmentOrder`—Creates one or more fulfillment orders and fulfillment order products for an order delivery group summary, which defines a recipient and delivery method. • `createFulfillmentOrders`—Creates fulfillment orders and fulfillment order products for multiple order delivery group summaries, each of which defines a recipient and delivery method. This value is available in API version 51.0 and later.

Field Name	Field Type	Description
		• `createInvoiceFromChangeOrders`—Creates an invoice for one or more change orders. This value is available in API version 56.0 and later. • `createInvoiceFromFulfillmentOrder`—Creates an invoice for a fulfillment order. • `createOrderPaymentSummary`—Creates an order payment summary for an authorization or payments belonging to an order summary. • `createOrderSummary`—Creates an order summary for an order. • `createReturnOrder`—Creates a return order and return order items for an order. • `ensureFundsOrderSummaryAsync`—Triggers an asynchronous background process to ensure funds through a payment provider for an invoice belonging to an order summary. • `ensureRefundsOrderSummaryAsync`—Triggers an asynchronous background process to ensure refunds through a payment provider for an invoice belonging to an order summary. • `getFulfillmentOrderCapacityValues`—Gets fulfillment order capacity information. This value is available in API version 55.0 and later. • `holdFulfillmentOrderCapacity`—Holds fulfillment order capacity. This value is available in API version 55.0 and later. • `orderRoutingFindRoutesWithFewestSplits`—Evaluates ordered product quantities against available inventory to determine the smallest combination of locations that can fulfill the order. This value is available in API version 51.0 and later. • `orderRoutingFindRoutesWithFewestSplitsUsingOCI`—Evaluates ordered product quantities against available inventory at specified location groups and locations to determine the smallest combination of locations that can fulfill the order. This value is available in API version 54.0 and later. • `orderRoutingRankByAverageDistance`—Calculates the average distance from sets of inventory locations to an order recipient, and returns the sets sorted by that average distance. This value is available in API version 51.0 and later. • `releaseHeldFulfillmentOrderCapacity`—Releases held fulfillment order capacity. This value is available in API version 55.0 and later. • `returnOrderItemSummariesPreview`—Previews the expected results of returning order item summaries from an order summary without actually returning them. • `returnOrderItemSummariesSubmit`—Returns order item summaries from an order summary. • `returnReturnOrderItems`—Processes return order line items. These values are used in Financial Services Cloud.

Field Name	Field Type	Description
		• `createFinancialRecords`—Creates person accounts, contacts, financial accounts, properties, assets, and liabilities from a residential loan application. This value is available in API version 49.0 and later. For values used in Fundraising for Nonprofit Cloud, see Flow for Fundraising. For values used in Health Cloud, see Flow for Health Cloud. For values used in Manufacturing Cloud, see Flow for Manufacturing Cloud. These values are used in Rebate Management. • `addRebateMemberList`—Adds a list of members to a rebate program. This value is available in API version 51.0 and later. • `calculateProjectedRebateAmount`—Calculates the projected rebate amount for rebate types associated with a specified transaction ID. This value is available in API version 54.0 and later. • `calculateRebateAmountAndUpsertPayout`—Calculates the rebate amount and upserts the rebate payout for the specified aggregate record. This value is available in API version 51.0 and later. • `getBenefitAndCalculateRebateAmount`— Gets benefit details, and optionally calculates the rebate amount for the specified aggregate record. This value is available in API version 51.0 and later. • `getEligibleProgramRebateTypes`—Retrieves the eligible program rebate types for a mapped object. This value is available in API version 52.0 and later. • `generateRebatePayoutPeriods`—Generates payout periods for a rebate program based on the frequency specified in the program. This value is available in API version 51.0 and later. • `processRebatesBatchCalculationJob`—Processes a rebate batch calculation job from the Data Processing Engine. This value is available in API version 51.0 and later. • `processProgramRebateTypeProducts`—Insert or delete records in the Program Rebate Type Product object. This value is available in API version 53.0 and later. • `rebatesProcessCSV`—Processes an uploaded CSV file using Bulk API 2.0 and converts the file's data into records in the target object. This value is available in API version 51.0 and later. • `upsertCustomRebatePayout`—Upserts the custom calculated rebate payout for the specified aggregate record. This value is available in API version 51.0 and later. These values are used in Loyalty Management. • `adjustPoints`—Adjusts loyalty points for a specified program member or journal transaction. This value is available in API version 51.0 and later. • `assignTierBenefits`— Assigns Member Benefits to a member tier for benefits that are associated with a Benefit Action. This value is available in API version 51.0 and later.

Field Name	Field Type	Description
		• `cancelAccrual`—Cancels a specific set of accrual transactions. • `creditPoints`—Credits loyalty points to a specified program member's balance. This value is available in API version 51.0 and later. • `cancelRedemption`—Reverts a specific set of redemption transactions. This value is available in API version 51.0 and later. • `changeTier`—Changes the tier for a specified program member. This value is available in API version 51.0 and later. • `changeTierWhenNoErrors`—Changes tier for a specified loyalty program member only when all the input parameters meet the criteria. This value is available in API version 51.0 and later. • `debitPoints`—Debits loyalty points to a specified program member's balance. This value is available in API version 51.0 and later. • `executeMemberBenefit`—Processes the benefit action associated with the benefit, which is assigned to a loyalty program member. This value is available in API version 51.0 and later. • `generateMemberReferralCode`—Generates a unique 8-character referral code for a loyalty program member. This value is available in API version 57.0 and later. • `getMemberActiveSegments`—Retrieve active Data Cloud market segments that a loyalty program member is a part of. • `getTier`—Gets the current tier for a specified program member. This value is available in API version 51.0 and later. • `getPointsBalance`—Gets the loyalty points balance for a specified program member. This value is available in API version 51.0 and later. • `getLoyaltyPromotion`—Gets active loyalty promotions based on a transaction journal. This value is available in API version 53.0 and later. • `getLoyaltyPromotionBasedOnSalesforceCDP`—Gets promotions for a member based on the market segment the member belongs to. This value is available in API version 53.0 and later. • `issueVoucher`—Issues a voucher for a member or contract. This value is available in API version 51.0 and later. • `mergeLoyaltyProgramMembership`—Merges two active loyalty program member records that both belong to the same loyalty program. This value is available in API version 56.0 and later. • `transferMemberPointsToGroups`—Transfers points from an individual member or a corporate member to the member's associated group. This value is available in API version 53.0 and later. • `updateProgressForCumulativePromotionUsage`—Updates the progress a member has made towards attaining a cumulative type promotion. This value is available in API version 53.0 and later. • `unmergeLoyaltyProgramMembership`—Unmerges loyalty program member records that have a Merged status. The action unmerges

Field Name	Field Type	Description
		memberships in the Merged status from the previously merged membership. This value is available in API version 56.0 and later. • `runProgramProcess`—Triggers an active loyalty program process. This value is available in API version 56.0 and later. • `runProgramProcessForTransactionJournal`—Triggers an active loyalty program process whose process type is TransactionJournal. This value is available in API version 54.0 and later. These values are for Decision Table. • `decisionTableAction`—Runs an active decision table definition. This value is available in API version 51.0 and later. • `refreshDecisionTable`—Refreshes the decision table cache. This value is available in API version 51.0 and later. These values are for the Batch Management jobs. • `batchJobAction`—Runs the batch management jobs definitions. This value is available in API version 51.0 and later. • `submitFailedRecordsBatchJob`—Resubmits an existing batch job with failed records for processing. This value is available in API version 52.0 and later. This value is for Data Processing Engine. • `dataProcessingEngineAction`—Runs the data processing engine definitions. This value is available in API version 51.0 and later. This value is used for Einstein Visit Recommendation. • `saveRecommendationDecision`—Save visit and task recommendation decisions. This value is available in API version 51.0 and later. These values are used in Field Service. If no version is specified, the value is available in API version 52.0 and later. • `addWorkPlans`—Creates work plan and work step objects from the work plan library. • `addWorkSteps`—Creates work step objects from the work plan library. • `deleteWorkPlans`—Deletes all the work plans and work steps associated with a work order or work order line item. • `generateWorkPlans`—Generates work plans based off rules defined in the work plan library. For values used in Intelligent Form Reader, see Flow for Intelligent Form Reader. For values used in Intelligent Document Reader, see Flow for Intelligent Document Reader. This value is used in Public Sector Solutions. • `createBenefitDisbursement`—Creates a benefit disbursement for an eligible benefit assignment. This value is available in API version 57.0 and later.

Field Name	Field Type	Description
		• `runRecordAggrBatchProcDef`—Runs a Data Processing Engine definition to process an asynchronous batch job that creates or updates record aggregation results. This value is available in API version 59.0 and later. These values are reserved for future use. • `thanks` • `metricRefresh` • `exportSurveyResponses`

StrategyNodeInvocableActionArg

Defines arguments passed to an Apex invocable action that generates or enhances a list of recommendations.

Field Name	Field Type	Description
`name`	string	Required. Unique name for the parameter to pass to the invocable action. The name must match a parameter that's defined in the invocable action.
`value`	string	Required. A Salesforce formula expression that is evaluated with the result used as the parameter value for the action.

StrategyNodeRecommendationLimit

Filters out recommendations that have already been accepted or rejected. Extends StrategyNodeUnionBase and inherits all of its fields.

Field Name	Field Type	Description
`filterMode`	StrategyReactionType (enumeration of type string)	Available reactions to filter out. The valid values are: • `Accepted` • `Rejected`
`lookbackDuration`	int	Number of days to search back.
`maxRecommendationCount`	int	Maximum number of times recommendation has been accepted or rejected.

StrategyNodeRecommendationLoad

Retrieves Recommendation objects. Extends StrategyNodeUnionBase and inherits all of its fields.

Field Name	Field Type	Description
`condition`	RecommendationLoadCondition[]	Array of conditions specifying which recommendations to load.

Field Name	Field Type	Description
`conditionLogic`	string	Logic to combine conditions, either AND or OR. All conditions are combined (not mixed). For example: `Cond1 AND Cond2 AND Cond3`.
`object`	string	Required. Specifies the API name of the sObject from which recommendations are loaded. For example, the field references `Account` or `MyCustomObject__c` and not a specific record of that object. Available in API version 48.0 and later.
`sortField`	StrategyNodeSortField	The field to sort on. Available in API version 48.0 and later.

RecommendationLoadCondition

Represents a condition used as part of the query constructed by StrategyNodeRecommendationLoad.

Field Name	Field Type	Description
`field`	string	Required. Any field from Recommendation BPO (SOAP) object.
`operator`	RecommendationConditionOperator (enumeration of type string)	Required. Valid values are: • `EQUALS` • `GREATER_THAN` • `GREATER_THAN_OR_EQUAL_TO` • `LESS_THAN` • `LESS_THAN_OR_EQUAL_TO` • `NOT_EQUALS` • `LIKE` • `STARTS_WITH` • `ENDS_WITH=` • `CONTAINS`
`value`	RecommendationConditionValue	Required. Constant value to use in query.

RecommendationConditionValue

Represents a value used as part of a RecommendationCondition.

Field Name	Field Type	Description
`type`	RecommendationConditionValueType (enumeration of type string)	Required. Valid values are: • `TEXT` • `NUMBER`

Field Name	Field Type	Description
		• BOOLEAN • DATE • DATE_TIME • TIME
value	string	Required. The constant value.

StrategyNodeSortField

Defines the field to sort on for StrategyNodeSort and StrategyNodeRecommendationLoad.

Field Name	Field Type	Description
name	string	Required. Name of the field to sort.
nullsFirst	boolean	If true, null values are sorted to the beginning of the list. Defaults to false.
order	SortOrder (enumeration of type string)	Order in which the list is sorted. Defaults to Asc. Valid values are: • Asc (ascending) • Desc (descending)

StrategyNodeSort

Sorts the recommendations. Extends StrategyNodeUnionBase and inherits all of its fields.

Field Name	Field Type	Description
field	StrategyNodeSortField	Required. Field to sort on.

StrategyNodeUnion

StrategyNodeUnion combines the output of all its child nodes. StrategyNodeUnion is a concrete implementation of StrategtNodeUnionBase and inherits all its fields.

StrategyNodeMap

Set recommendation fields with values. Extends StrategyNodeUnionBase and inherits all of its fields.

Field Name	Field Type	Description
mapExpression	MapExpression on page 1697[]	List of MaxExpressions.

StrategyNodeExclusive

Returns results from the first child node that has results and no other. Extends StrategyNodeUnionBase and inherits all its fields.

MapExpression

Sets the value for a recommendation field used by the strategy.

Field Name	Field Type	Description
`expression`	string	Required. A formula expression that results in a valid value supported by the data type specified in the `type` field.
`name`	string	Required. Recommendation field name that the expression sets the value for.
`type`	string	Required. The data type of the value resulting from the value in the `expression` field. Valid values are: • `BOOLEAN` • `CURRENCY` • `DATE` • `DOUBLE` • `DATE_TIME` • `INTEGER` • `LONG` • `PERCENT` • `TEXT` • `TIME`

Declarative Metadata Sample Definition

The following is an example of a RecommendationStrategy component that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<RecommendationStrategy xmlns="http://soap.sforce.com/2006/04/metadata">
    <contextRecordType>Asset</contextRecordType>
    <description>Hills Brothers Coffee strategy to handle machine down
incidents</description>
    <if>
        <childNode>IfNoEscaladeOrBetterSupport</childNode>
        <childNode>IfModel</childNode>
        <description>If Machine Down</description>
        <label>RootNode</label>
        <name>RootNode</name>
        <childNodeExpression>
            <childName>IfModel</childName>
            <expression>ISPICKVAL($Record.Status, &amp;quot;OutOfOrder&amp;quot;)</expression>
        </childNodeExpression>
```

```
        <childNodeExpression>
            <childName>IfNoEscaladeOrBetterSupport</childName>
            <expression>ISPICKVAL($Record.Status, &quot;OutOfOrder&quot;)</expression>
        </childNodeExpression>
        <onlyFirstMatch>false</onlyFirstMatch>
    </if>
    <if>
        <childNode>LoadEscalade</childNode>
        <description>If Customer does not have escalade support plan</description>
        <label>IfNoEscaladeOrBetterSupport</label>
        <name>IfNoEscaladeOrBetterSupport</name>
        <childNodeExpression>
            <childName>LoadEscalade</childName>
            <expression>NOT(ISPICKVAL($Record.Account.SLA__c, &quot;Gold&quot;) ||
ISPICKVAL($Record.Account.SLA__c, &quot;Platinum&quot;))</expression>
        </childNodeExpression>
        <onlyFirstMatch>false</onlyFirstMatch>
    </if>
    <if>
        <childNode>LoadMiniDiagnostic</childNode>
        <childNode>LoadMaxiDiagnostic</childNode>
        <description>If Machine Model switch node</description>
        <label>IfModel</label>
        <name>IfModel</name>
        <childNodeExpression>
            <childName>LoadMiniDiagnostic</childName>
          <expression>$Record.Product2.Name == &quot;Mini Coffee Roaster&quot;</expression>

        </childNodeExpression>
        <childNodeExpression>
            <childName>LoadMaxiDiagnostic</childName>
          <expression>$Record.Product2.Name == &quot;Maxi Coffee Roaster&quot;</expression>

        </childNodeExpression>
        <onlyFirstMatch>false</onlyFirstMatch>
    </if>
    <label>HillsBrothersCoffee</label>
    <recommendationLoad>
        <description>Load upgrade to escalade support plan</description>
        <label>LoadEscalade</label>
        <name>LoadEscalade</name>
        <condition>
            <field>Name</field>
            <operator>EQUALS</operator>
            <value>
                <type>TEXT</type>
                <value>Upgrade your Maintenance Package</value>
            </value>
        </condition>
        <conditionLogic>and</conditionLogic>
    </recommendationLoad>
    <recommendationLoad>
        <description>Load Mini Coffee Roaster Diagnostic Troubleshooting
proposition</description>
```

```
        <label>LoadMiniDiagnostic</label>
        <name>LoadMiniDiagnostic</name>
        <condition>
            <field>Name</field>
            <operator>EQUALS</operator>
            <value>
                <type>TEXT</type>
                <value>Mini Coffee Roaster Diagnostic Troubleshooting</value>
            </value>
        </condition>
        <conditionLogic>and</conditionLogic>
    </recommendationLoad>
    <recommendationLoad>
        <description>Load Maxi Coffee Roaster Diagnostic Troubleshooting
proposition</description>
        <label>LoadMaxiDiagnostic</label>
        <name>LoadMaxiDiagnostic</name>
        <condition>
            <field>Name</field>
            <operator>EQUALS</operator>
            <value>
                <type>TEXT</type>
                <value>Maxi Coffee Roaster Diagnostic Troubleshooting</value>
            </value>
        </condition>
        <conditionLogic>and</conditionLogic>
    </recommendationLoad>
    <union>
        <childNode>RootNode</childNode>
        <label>Output</label>
        <name>Output</name>
    </union>
    <invocableAction>
        <action>MyInvocableApexClass</action>
        <isGenerator>true</isGenerator>
        <type>apex</type>
        <argument>
            <name>MyNameParam</name>
            <value>$User.FirstName</value>
        </argument>
        <argument>
            <name>MyIdParam</name>
            <value>$Record.Id</value>
        </argument>
    </invocableAction>
    <map>
        <expression>
            <name>Name</name>
            <expression>'Hello' & $User.FirstName</expression>
            <type>TEXT</type>
        </expression>
        <expression>
            <name>MyDynamicField</name>
            <expression>Id == $Record.Id</expression>
```

```
            <type>BOOLEAN</type>
        </expression>
    </map>
</RecommendationStrategy>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

RecordActionDeployment

Represents configuration settings for the Actions & Recommendations, Action Launcher, and Bulk Action Panel components. For example, you can have a deployment that specifies which types of actions to display, default actions for channels, and the actions that users can add at runtime. If the component shows Next Best Action recommendations, the deployment configures which strategies to use and how recommendations appear. This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

RecordActionDeployment values are stored in the *developer_name*`.deployment` file in the `recordActionDeployments` directory.

Note: We don't recommend programmatically changing the API name of a RecordActionDeployment.

Version

RecordActionDeployment is available in API version 45.0 and later.

Fields

Field Name	Field Type	Description
`channelConfigurations`	RecordActionDeploymentChannel	Specifies configuration settings for different channels in an Actions & Recommendations deployment.
`componentName`	ComponentName (enumeration of type string)	Specifies the name of the component used in the deployment: • `ActionsAndRecommendations`—0 • `ActionLauncher`—1 • `BulkActionPanel`—2. This value is available in API version 60.0 and later For example, a value of 1 indicates that 1 is stored in the database if Action Launcher is used to create a deployment. Available in API version 56.0 and later.

Field Name	Field Type	Description
`deploymentContexts`	RecordActionDeploymentContext	Specifies the object context for quick actions and Next Best Action strategies. Available in API version 46.0 and later.
`hasComponents`	boolean	Indicates whether the record actions deployment includes components (`true`) or not (`false`). Available in API version 61.0 and later.
`hasGuidedActions`	boolean	Specifies that the component shows standard actions; for example, flows and quick actions. Available in API version 46.0 and later.
hasOmniscripts	boolean	Indicates whether the record actions deployment includes OmniScripts (`true`) or not (`false`). Available in API version 56.0 and later. The default value is `false`.
`hasRecommendations`	boolean	Specifies that the component shows recommendations from a Next Best Action strategy. Available in API version 46.0 and later.
`masterLabel`	string	Required. Specifies the name of the deployment.
`recommendation`	RecordActionRecommendation	Specifies settings for how Next Best Action recommendations appear in the component. Available in API version 46.0 and later.
`selectableItems`	RecordActionDeploymentSelectableItems	Specifies the actions that users can add at runtime.
`shouldLaunchActionOnReject`	boolean	Required. If `true`, launch the flow when the recommendation is rejected by the agent. Available in API version 48.0 and later.

RecordActionDefaultItem

Represents actions and attributes specified as channel defaults in a deployment.

Field Name	Field Type	Description
`action`	string	Required. Specifies the API name of an action. For example, the API name of a flow, such as `Verify_Information`.
`isMandatory`	boolean	Specifies whether the action is marked as mandatory. The default value is `false`.
`isUiRemoveHidden`	boolean	Specifies whether the remove option is hidden in the UI. The default value is false. If `true`, the UI hides the ability to remove the action from the list.
`pinned`	PinnedAction (enumeration of type string)	Required. Indicates whether the action is pinned to the `Top` or `Bottom`, or unpinned (`None`). The default value is `None`.
`position`	int	Required. Indicates the order of the action among all actions associated with this record.
`type`	RecordActionType (enumeration of type string)	Required. The type of action that's associated with the record. Valid values are: • `Flow` • `QuickAction` (Available in API version 46.0 and later.)

Field Name	Field Type	Description
		• `OmniScript` (Available in API version 56.0 and later.) • `LWC` (Available in API version 62.0 and later.) • `SvcCatalogItemDef` (Available in API version 62.0 and later.) • `WebLink` (Available in API version 62.0 and later.)

RecordActionDeploymentChannel

Specifies channel-specific defaults to show in the Actions & Recommendations component. The component displays the channel defaults when the list is otherwise empty.

Field Name	Field Type	Description
`channel`	ChannelSource (enumeration of type string)	Required. Specifies the channel. Valid values are `Phone`, `Chat`, or `Default`.
`channelItems`	RecordActionDefaultItem	Specifies default actions for a channel and attributes for each action, such as whether the action is pinned to the list top or bottom or whether an action is considered mandatory.
`isAutopopEnabled`	boolean	Specifies whether the first action in the list is launched when the record page opens. If `true`, the first action is launched. The default value is `false`.

RecordActionDeploymentContext

Specifies an object that provides context for quick actions and Next Best Action strategies. When the component appears on this type of page, it includes object-specific quick actions and uses an object-specific strategy to filter recommendations. Available in API version 46.0 and later.

Note: We support a maximum of 10 objects that provide context within a deployment.

Field Name	Field Type	Description
`entityName`	string	Required. Specifies the API name of an object to use as context.
`recommendationStrategy`	string	Specifies the API name of a Next Best Action strategy that overrides the default strategy on this page. A strategy is a metadata type RecommendationStrategy.

RecordActionRecommendation

Specifies settings to display Next Best Action recommendations in the component. Available in API version 46.0 and later.

Field Name	Field Type	Description
`defaultStrategy`	string	Specifies the API name of the default Next Best Action strategy, which is a metadata type, RecommendationStrategy.

Field Name	Field Type	Description
`hasDescription`	boolean	Required. If `true`, display the description for the recommendation.
`hasImage`	boolean	Required. If `true`, display the image for the recommendation.
`hasRejectAction`	boolean	Required. If `true`, display the label that the user clicks to reject the recommendation.
`hasTitle`	boolean	Required. If `true`, display the title for the recommendation.
`maxDisplayRecommendations`	int	Required. Specifies the maximum number of recommendations to display. Valid values are 1–4.

RecordActionSelectableItem

Represents the set of actions available for users to add to the component at runtime.

Field Name	Field Type	Description
`action`	string	Required. Specifies the API name of an action. For example, the API name of a flow, such as `Verify_Information`.
`type`	RecordActionType (enumeration of type string)	Required. The type of action that's associated with the record. Valid values are: • `Flow` • `QuickAction` (Available in API version 46.0 and later.) • `OmniScript` (Available in API version 56.0 and later.) • `LWC` (Available in API version 62.0 and later.) • `SvcCatalogItemDef` (Available in API version 62.0 and later.) • `WebLink` (Available in API version 62.0 and later.)
`isFrequentAction`	boolean	Indicates whether an action is frequently accessed by users (`true`) or not (`false`). Available in version 57.0 and later. This field applies only to Action Launcher.
frequentActionSequenceNbr	integer	The sequence number that's assigned to a frequently used action that's shown on Action Launcher. Available in version 57.0 and later. This field applies only to Action Launcher.

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

Declarative Metadata Sample Definition

The following is a sample of a `recordActionDeployment` file.

```
<RecordActionDeployment xmlns="http://soap.sforce.com/2006/04/metadata">
    <channelConfigurations>
        <channel>Phone</channel>
        <channelItems>
            <action>Sample_Flow</action>
            <isMandatory>false</isMandatory>
            <isUiRemoveHidden>false</isUiRemoveHidden>
            <position>1</position>
            <pinned>Top</pinned>
            <type>Flow</type>
        </channelItems>
        <channelItems>
            <action>Another_Sample_Flow</action>
            <isMandatory>false</isMandatory>
            <isUiRemoveHidden>true</isUiRemoveHidden>
            <position>2</position>
            <pinned>Top</pinned>
            <type>Flow</type>
        </channelItems>
        <isAutopopEnabled>true</isAutopopEnabled>
    </channelConfigurations>
    <masterLabel>Sample Deployment</masterLabel>
    <selectableItems>
        <action>Sample_Flow</action>
        <type>Flow</type>
        <isFrequentAction>true</isFrequentAction>
        <frequentActionSequenceNbr>1</frequentActionSequenceNbr>
    </selectableItems>
    <selectableItems>
        <action>Sample_Flow_2</action>
        <type>Flow</type>
        <isFrequentAction>false</isFrequentAction>
    </selectableItems>
    <hasGuidedActions>true</hasGuidedActions>
    <hasRecommendations>true</hasRecommendations>
    <recommendation>
        <defaultStrategy>Sample_Global_Strategy</defaultStrategy>
        <maxDisplayRecommendations>4</maxDisplayRecommendations>
        <hasImage>true</hasImage>
        <hasDescription>true</hasDescription>
        <hasRejectAction>true</hasRejectAction>
        <hasTitle>true</hasTitle>
    </recommendation>
    <deploymentContexts>
        <entityName>Case</entityName>
        <recommendationStrategy>Sample_Case_Strategy</recommendationStrategy>
    </deploymentContexts>
    <deploymentContexts>
        <entityName>Account</entityName>
        <recommendationStrategy>Sample_Acc_Strategy</recommendationStrategy>
```

```
    </deploymentContexts>
</RecordActionDeployment>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <fullName>SecondTest</fullName>
    <types>
        <members>Sample_Flow</members>
        <members>Another_Sample_Flow</members>
        <members>Sample_Flow_2</members>
        <name>Flow</name>
    </types>
    <types>
        <members>SampleDeployment</members>
        <name>RecordActionDeployment</name>
    </types>
    <version>45.0</version>
</Package>
```

SEE ALSO:

RecommendationStrategy

RecordAggregationDefinition

Represents a data aggregation from one object to another object to which it is connected by other objects in the data model.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

RecordAggregationDefinition components have the suffix `.RecordAggregationDefinition` and are stored in the `RecordAggregationDefinitions` folder.

Version

RecordAggregationDefinition components are available in API version 59.0 and later.

Special Access Rules

To access the RecordAggregationDefinition metadata type, you must have the Record Aggregation permission set license and the Record Aggregation Access permission.

Fields

Field Name	Description
`aggregateFromObject`	**Field Type** string **Description** Required. API name of the object from which data is aggregated.
`aggregateToObject`	**Field Type** string **Description** Required. API name of the object to which data is aggregated.
`aggregationType`	**Field Type** RecordAggregationDefinitionAggregationType (enumeration of type string) **Description** Required. Type of the data aggregation. Valid value is: • `Record`
`batchProcessingDefinition`	**Field Type** string **Description** Data Processing Engine definition that aggregates data from one record to another.
`description`	**Field Type** string **Description** Description for this record aggregation definition.
`displayName`	**Field Type** string **Description** Required. Name of the record aggregation definition that's displayed in the record page.
`recordAggregationObject`	**Field Type** RecordAggregationObject[]

Field Name	Description
	Description List of record aggregation objects in the record aggregation join sequence.
`status`	**Field Type** RecordAggregationDefinitionStatus (enumeration of type string) **Description** Required. Status of this record aggregation definition. Values are: • `Active` • `Draft` • `Inactive`

RecordAggregationObject

Represents an object in the record aggregation join sequence.

Field Name	Description
`associatedObject`	**Field Type** string **Description** Required. API name of the object associated with this record aggregation object.
`developerName`	**Field Type** string **Description** Developer name of the record aggregation object. May contain only underscores and alphanumeric characters and must be unique in your org. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
`filterLogic`	**Field Type** string **Description** Logical sequence in which the record aggregation object filters associated with this record aggregation object are applied to the associated object's records. If you define two or more record aggregation object filters, but don't specify the sequence in which to apply the filters, the filters are applied by using a logical AND expression. Available in API version 60.0 and later.

Field Name	Description
masterLabel	**Field Type** string **Description** Required. A user-friendly name for RecordAggregationDefinition, which is defined when the RecordAggregationDefinition is created.
recordAggregationJoinCondition	**Field Type** RecordAggregationJoinCondition[] **Description** List of join conditions that apply to this record aggregation object.
recordAggregationObjectFilter	**Field Type** RecordAggregationObjectFilter[] **Description** List of filters that are applied to the records of this record aggregation object. Available in API version 60.0 and later.

RecordAggregationJoinCondition

Represents a condition in a join between two record aggregation objects.

Field Name	Description
joinField	**Field Type** string **Description** Required. API name of the field on the record aggregation object's associated object that is used in the join condition.
navigationSequenceNumber	**Field Type** int **Description** Required. Sequence number corresponding to this join in the join sequence from the object to which the data is aggregated to the object that contains the data being aggregated.
relatedJoinField	**Field Type** string

Field Name	Description
	Description Required. API name of the field on the related record aggregation object's associated object that is used in the join condition.
`relatedRecordAggregationObject`	**Field Type** string **Description** Required. Second record aggregation object in the join condition.
`type`	**Field Type** RecordAggregationJoinConditionType (enumeration of type string) **Description** Required. Type of this record aggregation join in the join path from the object to which the data is aggregated to the object that contains the data being aggregated. Valid values are: • `AggregateFrom` • `AggregateTo` • `Intermediate`

RecordAggregationObjectFilter

Represents a filter that is applied to the records of an object in the record aggregation join sequence. Available in API version 60.0 and later.

Field Name	Description
`associatedObjectField`	**Field Type** string **Description** Required. API name of the associated object's field whose value is used to filter the object's records. The associated object is specified in the record aggregation object.
`operator`	**Field Type** RecordAggregationObjectFilterOperator (enumeration of type string) **Description** Required.

Field Name	Description
	Operator used in the filter expression. Values are: • Contains • Equals • GreaterThan • GreaterThanOrEquals • In • LessThan • LessThanOrEquals • NotEquals • NotIn
sequenceNumber	**Field Type** int **Description** Required. Sequence number of this record aggregation object filter.
value	**Field Type** string **Description** Required. Reference value with which the designated field's values are compared when the filter is applied on the associated object's records.

Declarative Metadata Sample Definition

The following is an example of a RecordAggregationDefinition component.

```
<?xml version="1.0" encoding="UTF-8"?>
<RecordAggregationDefinition xmlns="http://soap.sforce.com/2006/04/metadata">
    <aggregateToObject>PartyRelationshipGroup</aggregateToObject>
    <aggregateFromObject>PartyIncome</aggregateFromObject>
    <status>Active</status>
    <aggregationType>Record</aggregationType>
    <description>Aggregate head of household's income to household</description>
    <displayName>Party Income to Party Relationship Group</displayName>
    <recordAggregationObject>
        <associatedObject>PartyRelationshipGroup</associatedObject>
        <masterLabel>Party Relationship Group Object</masterLabel>
        <developerName>PartyRelationshipGroupObject</developerName>
        <recordAggregationJoinCondition>
```

```
            <joinField>Account</joinField>
            <navigationSequenceNumber>1</navigationSequenceNumber>
            <relatedJoinField>Account</relatedJoinField>

<relatedRecordAggregationObject>AccountContactrelationObject</relatedRecordAggregationObject>

            <type>Intermediate</type>
        </recordAggregationJoinCondition>
        <recordAggregationObjectFilter>
          <associatedObjectField>Type</associatedObjectField>
          <operator>Equals</operator>
          <value>Household</value>
          <sequenceNumber>1</sequenceNumber>
        </recordAggregationObjectFilter>
    </recordAggregationObject>
    <recordAggregationObject>
        <associatedObject>AccountContactRelation</associatedObject>
        <masterLabel>Account Contact Relation Object</masterLabel>
        <developerName>AccountContactRelationObject</developerName>
        <recordAggregationJoinCondition>
            <joinField>Contact</joinField>
            <navigationSequenceNumber>2</navigationSequenceNumber>
            <relatedJoinField>Party</relatedJoinField>

<relatedRecordAggregationObject>PartyIncomeObject</relatedRecordAggregationObject>
            <type>Intermediate</type>
        </recordAggregationJoinCondition>
        <recordAggregationObjectFilter>
          <associatedObjectField>IsPrimaryMember</associatedObjectField>
          <operator>Equals</operator>
          <value>true</value>
          <sequenceNumber>1</sequenceNumber>
        </recordAggregationObjectFilter>
    </recordAggregationObject>
    <recordAggregationObject>
        <associatedObject>PartyIncome</associatedObject>
        <masterLabel>Party Income Object</masterLabel>
        <developerName>PartyIncomeObject</developerName>
        <filterLogic>1 AND 2</filterLogic>
        <recordAggregationObjectFilter>
          <associatedObjectField>IncomeFrequency</associatedObjectField>
          <operator>Equals</operator>
          <value>Monthly</value>
          <sequenceNumber>1</sequenceNumber>
        </recordAggregationObjectFilter>
        <recordAggregationObjectFilter>
          <associatedObjectField>IncomeStatus</associatedObjectField>
          <operator>Equals</operator>
          <value>Active</value>
          <sequenceNumber>2</sequenceNumber>
        </recordAggregationObjectFilter>
    </recordAggregationObject>
</RecordAggregationDefinition>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>RecordAggregationDefinition</name>
    </types>
    <version>60.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

RecordAlertCategory

Represents a category to group and present record alerts.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

RecordAlertCategory components have the suffix `recordAlertCategory` and are stored in the `recordAlertCategories` folder.

Version

RecordAlertCategory components are available in API version 54.0 and later.

Fields

Field Name	Description
`description`	**Field Type** string **Description** The description of the record alert category.
`masterLabel`	**Field Type** string

Field Name	Description
	Description Required. The user-interface name of the record alert category.
`severity`	**Field Type** string **Description** Indicates the degree of impact that an alert in this category can have. Possible Education Cloud values are: • `High` • `Low` • `Medium` Possible Financial Service Cloud values are: • `Error` • `Info` • `Minor` • `Warning`

Declarative Metadata Sample Definition

The following is an example of a RecordAlertCategory component.

```
<?xml version="1.0" encoding="UTF-8"?>
<RecordAlertCategory xmlns="http://soap.sforce.com/2006/04/metadata">
    <description>Tracks Financial Account Fraud Alerts</description>
    <masterLabel>Fraud</masterLabel>
    <severity>Error</severity>
</RecordAlertCategory>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>Fraud</members>
        <name>RecordAlertCategory</name>
    </types>
    <version>54.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

RegisteredExternalService

Represents a registered external service, which provides an extension or integration.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

RegisteredExternalService components have the suffix `.registeredExternalService` and are stored in the `registeredExternalServices` folder.

Version

RegisteredExternalService components are available in API version 49.0 and later.

Special Access Rules

This metadata type is available only if the B2B Commerce or D2C Commerce license is enabled.

Fields

Field Name	Description
`configUrl`	**Field Type** string **Description** Link to the configuration page for the integration.
`description`	**Field Type** string **Description** Description of the external service provider. This field is available in API version 59.0 and later.
`documentationUrl`	**Field Type** string

Field Name	Description
	Description Link to documentation for the registered external service.
`extensionPointName`	**Field Type** ExtensionPointName (enumeration of type string) **Description** This field is available in API version 55.0 and later. Name of an extension point. Possible values are: • `Commerce_Domain_Cart_Calculate` • `Commerce_Domain_Checkout_CreateOrder` • `Commerce_Domain_Inventory_CartCalculator` • `Commerce_Domain_Inventory_Service` • `Commerce_Domain_OrderManagement_Product` • `Commerce_Domain_Pricing_CartCalculator` • `Commerce_Domain_Pricing_Service` • `Commerce_Domain_Promotions_CartCalculator` • `Commerce_Domain_Promotions_ShippingCalculator` • `Commerce_Domain_Shipping_CartCalculator` • `Commerce_Domain_Shipping_SplitShipment` • `Commerce_Domain_Tax_CartCalculator` • `Commerce_Domain_Tax_Service` • `Commerce_Endpoint_Account_Address` • `Commerce_Endpoint_Account_Addresses` • `Commerce_Endpoint_Cart_Item`—This field value is available in API version 62.0 and later. • `Commerce_Endpoint_Cart_ItemCollection`—This field value is available in API version 62.0 and later. • `Commerce_Endpoint_Catalog_Product` • `Commerce_Endpoint_Catalog_Products` • `Commerce_Endpoint_Search_ProductSearch` • `Commerce_Endpoint_Search_Products` • `Commerce_Endpoint_Search_ProductsByCategory`
`externalServiceProvider`	**Field Type** string **Description** Required. The ID of an Apex class functioning as a provider. The Apex class can either implement one of the following interfaces: • sfdc_checkout.CartInventoryValidation

Field Name	Description
	• sfdc_checkout.CartPriceCalculations • sfdc_checkout.CartShippingCharges • sfdc_checkout.CartTaxCalculations or the Apex class can extend one of the base classes for an extension. See Available Extensions.
`externalServiceProviderType`	**Field Type** RegistryProviderType (enumeration of type string) **Description** Required. The type of external service provider. For an extension, you set the type to `Extension`, and you specify an `extensionPointName`. For example, for a Pricing Cart Calculator extension, you specify `Commerce_Domain_Pricing_CartCalculator` as the `extensionPointName`. For an integration, you set the type to one of the other possible values, such as `Price`, and you omit `extensionPointName`. Possible values are: • `Extension` (this value is available in API version 55.0 and later) • `Inventory` • `Price` • `Promotions` (this value is available in API version 53.0 and later) • `Shipment` • `Tax`
`iconUri`	**Field Type** string **Description** URI of icon for the extension provider. This field is available in API version 59.0 and later.
`isApplication`	**Field Type** boolean **Description** Indicates if the extension provider is contained within a managed package. The default value is `false`. This field is available in API version 59.0 and later.
`isProtected`	**Field Type** boolean **Description** An auto-generated value that doesn't impact the behavior of the metadata type. The default value is `false`.

Field Name	Description
masterLabel	**Field Type** string **Description** Required. The primary label for the RegisteredExternalService.

Declarative Metadata Sample Definition

The following is an example of a RegisteredExternalService component.

```
<?xml version="1.0" encoding="UTF-8"?>
<RegisteredExternalService xmlns="http://soap.sforce.com/2006/04/metadata">
    <externalServiceProvider>TaxSample</externalServiceProvider>
    <externalServiceProviderType>Tax</externalServiceProviderType>
    <documentationUrl>http://sample.com/doc</documentationUrl>
    <configUrl>http://sample.com/config</configUrl>
    <masterLabel>TaxService</masterLabel>
    <isProtected>false</isProtected>
</RegisteredExternalService>
```

The following is an example `package.xml` that references the previous definition.

```
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>TaxSample</members>
        <name>ApexClass</name>
    </types>
    <types>
        <members>TaxService</members>
        <name>RegisteredExternalService</name>
    </types>
    <version>60.0</version>
</Package>
```

ReferencedDashboard

Represents the ReferencedDashboard object in CRM Analytics. A referenced dashboard stores information about an externally referenced dashboard.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

`ReferencedDashboard` components have the suffix `.refdash` and are stored in the `wave` folder.

Version

ReferencedDashboard components are available in API version 57.0 and later.

Special Access Rules

To view referenced dashboards, you need the Enables Tableau Dashboards in CRM Analytics permission.

Fields

Field Name	Field Type	Description
`application`	string	Required. The internal name of the Analytics app.
`description`	string	The dashboard description that appears in the user interface.
`embedUrl`	string	Required. The URL to the referenced dashboard.
`masterLabel`	string	Required. The dashboard name that appears in the user interface.
`templateAssetSourceName`	string	Links the dashboard to the template used to create it. Null for assets not created from a template.
`visibility`	string	The visibility of the dashboard. Valid values are: `ALL` and `LIMITED`.

Declarative Metadata Sample Definition

The following is an example of a WaveDashboard component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ReferencedDashboard xmlns="http://soap.sforce.com/2006/04/metadata"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <application>my_app</application>
    <masterLabel>ReferencedDashboard1</masterLabel>
    <description>My Tableau Dashboard</description>
    <embedUrl>https://public.tableau.com/views/Superstore_24/Overview</embedUrl>
    <visibility>ALL</visibility>
</ReferencedDashboard>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

RelatedRecordAssocCriteria

Represents criteria for automatically linking records like accounts, leads, opportunities, and cases with the branches that work with them.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

RelatedRecordAssocCriteria components have the suffix `.relatedRecordAssocCriteria` and are stored in the `relatedRecordAssocCriteria` folder.

Version

RelatedRecordAssocCriteria components are available in API version 52.0 and later.

Special Access Rules

To use this object, you must have the Financial Services Cloud Extension permission set.

Fields

Field Name	Description
`associationHandlerApexClass`	**Field Type** string **Description** The name of a custom Apex class that handles the creation of association records for specific association criteria. This class must: • Apply to an object that the Record Association Builder doesn't directly support • Implement the `fscwmgen.BranchManagement AssociationHandler` interface • Return a list of Branch Unit Related Records • Populate at least the minimum required fields in each Branch Unit Related Record: – `BranchUnitId`: Represents the current branch unit of the user or contact – `BusinessUnitMemberId`: The Banker ID of the user or contact – `RelatedRecordId`: The ID of the custom object to be related This field is a relationship field.

Field Name	Description
`associationType`	**Field Type** AssociationType (enumeration of type string) **Description** Required. The association type. Values are: • `BranchManagement`
`description`	**Field Type** string **Description** A description of the association criteria.
`eventType`	**Field Type** AssociationEventType (enumeration of type string) **Description** Required. The type of reference object event that triggers creation of the association. Values are: • `Create` • `Update`
`isProtected`	**Field Type** boolean **Description** An auto-generated value that doesn't impact the behavior of the metadata type. The default value is `false`.
`masterLabel`	**Field Type** string **Description** Required. The master label of the association criteria. This internal label doesn't get translated.
`preCondition`	**Field Type** string **Description** Required. A formula that, when true, causes a new association to be created.
`referenceObject`	**Field Type** string

Field Name	Description
	Description Required. The reference object for the association criteria.
selectedOwnerField	**Field Type** string **Description** A field to use instead of the default Owner ID.
status	**Field Type** AssociationStatusType (enumeration of type string) **Description** Required. The status of the association criteria. Values are: • Active • Draft • Inactive

Declarative Metadata Sample Definition

The following is an example of a RelatedRecordAssocCriteria component.

```
<?xml version="1.0" encoding="UTF-8"?>
<RelatedRecordAssocCriteria xmlns="http://soap.sforce.com/2006/04/metadata">
   <associationType>BranchManagement</associationType>
   <eventType>Create</eventType>
   <masterLabel>RevenueThreeMillion</masterLabel>
   <preCondition>[Account].AnnualRevenue > 3000000</preCondition>
   <referenceObject>Account</referenceObject>
   <status>Active</status>
</RelatedRecordAssocCriteria>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>RelatedRecordAssocCriteria</name>
    </types>
    <version>52.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

RelationshipGraphDefinition

Represents a definition of a graph that you can configure in your organization to traverse object hierarchies and record details, giving you a glimpse of how your business works.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

RelationshipGraphDefinition components have the suffix `.relationshipGraphDefinition` and are stored in the `relationshipGraphDefinitions` folder.

Version

RelationshipGraphDefinition components are available in API version 55.0 and later.

Special Access Rules

The Financial Services Cloud permission set license is required to access this object.

Fields

Field Name	Description
`isActive`	**Field Type** boolean **Description** Required. Indicates whether the relationship graph is available for use (`true`) or not (`false`). The default value is `true`. **Note:** This field is read-only in API version 55.0.
`isTemplate`	**Field Type** boolean

Field Name	Description
	Description Required. Indicates whether you can configure this relationship graph as a template (`true` or not `false`). The default value is `false`. In the UI, this field is *Set as Template*.
`masterLabel`	**Field Type** string **Description** Required. A user-friendly name for RelationshipGraphDefinition, which is defined when the RelationshipGraphDefinition is created. In the UI, this field is *Label*.
`relationshipGraphDefVersions`	**Field Type** RelationshipGraphDefVersion[] **Description** Represents a list of graph versions associated with the relationship graph definition.

RelationshipGraphDefVersion

The list of graph versions associated with the relationship graph definition.

Field Name	Description
`graphDefinition`	**Field Type** string **Description** Required. Specifies a set of properties required to create a relationship graph, such as parent node, child relationships, filter and sort fields, and graph UI elements.
`graphType`	**Field Type** string **Description** Required. Specifies the type of graph. In API version 55.0, only `HorizontalHierarchy` graph type is supported.

Declarative Metadata Sample Definition

The following is an example of a RelationshipGraphDefinition component.

```
<?xml version="1.0" encoding="UTF-8"?>
<RelationshipGraphDefinition xmlns="http://soap.sforce.com/2006/04/metadata">
     <isActive>false</isActive>
     <isTemplate>true</isTemplate>
     <masterLabel>Account Graph</masterLabel>
     <relationshipGraphDefVersions>
         <graphDefinition>{
   "graph" : {
     "rootNode" : {
       "object" : {
         "entity" : "Account"
       },
       "configurationType" : "Primary",
       "sortFields" : [ {
         "field" : {
           "field" : "LastModifiedDate",
           "whichEntity" : "TARGET"
         },
         "order" : "DESC"
       } ],
       "nodeUiConfig" : {
         "fieldsToDisplay" : [ ],
         "showFieldLabels" : true,
         "actions" : { }
       },
       "childRelationships" : [ {
         "OneToMany" : {
           "targetObjectNode" : {
             "object" : {
               "entity" : "Contact"
             },
             "configurationType" :"Custom",
             "sortFields" : [ {
               "field" : {
                 "field" : "LastModifiedDate",
                 "whichEntity" : "TARGET"
               },
               "order" : "DESC"
             } ],
             "nodeUiConfig" : {
               "fieldsToDisplay" : [ {
                 "field" : "Name",
                 "whichEntity" : "TARGET"
               }, {
                 "field" : "Phone",
                 "whichEntity" :"TARGET"
               } ],
               "showFieldLabels" : true,
               "actions" : {
                 "containerActions" : [ {
                   "action" : "New"
```

```
                } ],
                "recordActions" : [ {
                  "action" : "Edit"
                }, {
                  "action" : "Delete"
                } ]
              }
            },
            "childRelationships" : [ ]
          },
          "relationshipUiConfig" : { },
          "filter" : {
            "filterCriteria" : [ {
              "field" : {
                "field" : "Name",
                "whichEntity" : "TARGET"
              },
              "operator" : "eq",
              "value" : "Salesforce"
            } ],
            "booleanFilter" : "1"
          },
          "targetObjectField" : {
            "field" : "AccountId",
            "whichEntity" : "TARGET"
          }
        }
      } ]
    },
    "globalUiConfig" : {
      "borderColor" : "Green2",
      "borderThickness" : "2px";,
      "colorShading" : "Black",
      "fieldLayout" : "Vertically Stacked",
      "recordContainerExpansion" : true,
      "recordExpansion" : true
    }
  }
 }</graphDefinition>
        <graphType>HorizontalHierarchy</graphType>
    </relationshipGraphDefVersions>
</RelationshipGraphDefinition>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
   <fullName>Package1</fullName>
   <types>
       <members>*</members>
       <name>RelationshipGraphDefinition</name>
   </types>
   <version>55.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

RemoteSiteSetting

Represents a remote site setting. Before any Visualforce page, Apex callout, or JavaScript code using XmlHttpRequest in an s-control or custom button can call an external site, that site must be registered in the Remote Site Settings page, or the call fails.

RemoteSiteSetting on page 1726 extends the Metadata metadata type and inherits its `fullName` field.

Declarative Metadata File Suffix and Directory Location

RemoteSiteSetting on page 1726 components are stored in the `remoteSiteSettings` directory of the corresponding package directory. The file name matches the unique name of the remote site setting, and the extension is `.remoteSite`.

Version

RemoteSiteSetting on page 1726 components are available in API version 19.0 and later.

Fields

Field	Field Type	Description
`description`	string	The description explaining what this remote site setting is used for.
`disableProtocolSecurity`	boolean	Required. Indicates whether code within Salesforce can access the remote site regardless of whether the user's connection is over HTTP or HTTPS (`true`) or not (`false`). When `true`, code within Salesforce can pass data from an HTTPS session to an HTTP session, and vice versa. Only set to `true` if you understand the security implications.
`fullName`	string	The name can only contain characters, letters, and the underscore (_) character. The name must start with a letter, and can't end with an underscore or contain two consecutive underscore characters. Inherited from the Metadata component, this field isn't defined in the WSDL for this component. It must be specified when creating, updating, or deleting. See create() to see an example of this field specified for a call.
`isActive`	boolean	Required. Indicates if the remote site setting is active (`true`) or not (`false`).
`url`	string	Required. The URL for the remote site.

Declarative Metadata Sample Definition

A sample XML definition of a remote site setting is shown in this code block.

```
<?xml version="1.0" encoding="UTF-8"?>
<RemoteSiteSetting xmlns="http://soap.sforce.com/2006/04/metadata">
    <description>Used for Apex callout to mapping web service</description>
    <disableProtocolSecurity>false</disableProtocolSecurity>
    <isActive>true</isActive>
    <url>https://www.maptestsite.net/mapping1</url>
</RemoteSiteSetting>
```

Report

Represents a custom report. This metadata type only supports custom reports; standard reports aren't supported.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

This type extends the Metadata metadata type and inherits its `fullName` field.

Declarative Metadata File Suffix and Directory Location

Reports are stored in the `reports` directory of the package directory. The file name consists of the report title with the extension `.report`.

Retrieving Reports

You can't use the wildcard (*) symbol with reports in `package.xml`.

To retrieve the list of explicit report names to populate `package.xml` with, first call `listMetadata(ListMetadataQuery[])` with a `ListMetadataQuery` entry with the `type` field set to `ReportFolder` and the `folder` field to `*` (wildcard). This call returns an array of FileProperties objects with the names of report folders in the `fullName` field.

Now call `listMetadata` with `ListMetadataQuery` entries where the `type` field is Report and the `folder` fields are the full name values from the first `listMetadata` call. These calls return `FileProperties` objects where the `fullName` field is the combination of the folder name and report name. Use these values in the `package.xml` to designate the members for the Report metadata type.

ReportFolder isn't returned as a type in `describeMetadata()`. Report is returned from `describeMetadata()` with an associated attribute of `inFolder` set to true. If that attribute is set to true, you can construct the type by using the component name with the word Folder, such as ReportFolder.

The following example shows folders in `package.xml`:

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>MyDBFolder/MyDBName</members>
        <name>Dashboard</name>
    </types>
    <types>
        <members>MyDocumentFolder/MyDocumentName</members>
```

```
        <name>Document</name>
    </types>
    <types>
        <members>unfiled$public/MarketingProductInquiryResponse</members>
        <members>unfiled$public/SalesNewCustomerEmail</members>
        <name>EmailTemplate</name>
    </types>
    <types>
        <members>MyReportFolder/MyReportName</members>
        <name>Report</name>
    </types>
    <version>64.0</version>
</Package>
```

Version

Report components are available in API version 14.0 and later.

Fields

The following information assumes that you're familiar with creating and running reports. For more information on these fields, see Build a Report in Salesforce Help.

Field	Field Type	Description
aggregateFilters	ReportAggregateFilter[]	List that defines filters on custom summary formulas for summary, matrix, and joined reports. Available in API version 64.0 and later.
aggregates	ReportAggregate[]	List that defines custom summary formulas for summary, matrix, and joined reports.
block	Report[]	Represents each block in a joined report where every block can be of a different report type.
blockInfo	ReportBlockInfo	Defines attributes for each block in a joined report.
buckets	ReportBucketField[]	Defines a bucket field to be used in the report. This field is available in API version 24.0 and later.
chart	ReportChart	Defines a chart for summary and matrix reports.
colorRanges	ReportColorRange[]	List that specifies conditional highlighting for report summary data. Salesforce Classic only.
columns	ReportColumn[]	List that specifies the fields displayed in the report. Fields appear in the report in the same order as they appear in the Metadata API file.

Field	Field Type	Description
`crossFilters`	ReportCrossFilter[]	Defines a cross filter's object, related object, and condition (WITH or WITHOUT). This field is available in API version 64.0 and later.
`currency`	`CurrencyIsoCode` (enumeration of type string)	When using multiple currencies, some reports allow you to display converted amounts by selecting the appropriate column to display. For example, in opportunity reports, you can include the Amount (converted) column on the report. This field is an enumeration of type string that defines the currency in which to display converted amounts. Valid values: Must be one of the valid alphabetic, three-letter currency ISO codes defined by the ISO 4217 standard, such as `USD`, `GBPSLE`, or `JPY`.
`dataCategoryFilters`	string	Specifies a filter according to the data category.
`description`	string	Specifies a general description, which is displayed with the report name. Maximum characters: 255 characters.
`division`	string	If your organization uses divisions to segment data and the Affected by Divisions permission is enabled, records in the report must match this division. This field is available in API version 17.0 and later.

`filter`	ReportFilter	Limits report results to records with specific data. For example, you can limit report results to opportunities for which the amount is greater than $1,000:
		`<filter>`
		`<criteriaItems>`
		`<column>AMOUNT</column>`
		`<operator>greaterThan</operator>`
		`<value>1000</value>`
		`</criteriaItems>`
		`</filter>`

`folderName`	string	Name of the folder that houses the report. This field is available in API version 35.0 and later.
`format`	ReportFormat (enumeration of type string)	Defines the report format. For example, `Tabular` for a simple data list without subtotals.

Field	Field Type	Description
formattingRules	ReportFormattingRule[] (enumeration of type string)	List that specifies conditional highlighting for report data. Lightning Experience only.
groupingsAcross	ReportGrouping[]	List that defines the fields by which you want to group and subtotal data across a matrix report (row headings). When grouping by a date field, you can further group the data by a specific time period such as days, weeks, or months. Maximum: 2 fields.
groupingsDown	ReportGrouping[]	For Summary and Matrix reports: List that defines the fields by which you want to group and subtotal. For summary reports, choosing more than one sort field allows you to subsort your data. For matrix reports, specifies summary fields for column headings. When grouping by a date field, you can further group the data by a specific time period such as days, weeks, or months. Maximum for matrix reports: 2. Maximum for summary reports: 3
historicalSelector	ReportHistoricalSelector	Defines a date range for which historical trend reporting data is to be captured. Default is "Any Historical Date." Available in API version 29.0 and later.
isSmartTotalDisabled	boolean	`false` displays smart totalling on the report. Available in API version 29.0 and later.
name	string	Required. The report name. For example, `Opportunity Pipeline`
numSubscriptions	int	Indicates whether a user has subscribed to this report Lightning Experience (1) or not (0). Tied to user context. This field is available in API version 38.0 and later.
params	ReportParam[]	List that specifies settings specific to each report type, in particular options that let you filter a report to obtain useful subsets. For example, the Activities report type lets you specify whether you want to see open or closed activities or both and whether you want to see tasks or events or both. Valid values depend on the report type.
reportCustomDetailFormula	CustomDetailFormulas	Allows you to apply row-level formulas to reports.

Field	Field Type	Description
`reportType`	string	Required. Defines the type of data in the report. For example, `Opportunity` to create a report of opportunities data.
`reportTypeApiName`	string	Defines the API Name for the report type. This field is available in API version 48.0 and later.
`roleHierarchyFilter`	string	The role name for a report drill down. Some reports, such as opportunity and activity reports, display Hierarchy links that allow you to drill down to different datasets based on the role hierarchy. This field is available in API version 17.0 and later.
`rowLimit`	int	Defines the maximum number of rows that can be returned for the report.
`scope`	string	Defines the scope of data on which you run the report. For example, whether you want to run the report against all opportunities, opportunities you own, or opportunities your team owns. Valid values depend on the `reportType`. For example, for Accounts reports: • `MyAccounts` • `MyTeamsAccounts` • `AllAccounts`
`showCurrentDate`	boolean	Can be set to `true` for historical trending reports in matrix format. Available in API version 29.0 and later.
`showDetails`	boolean	`false` shows a collapsed view of the report with only the headings, subtotals, and total. Default: `true`
`showGrandTotal`	boolean	`true` displays the calculated total for the full report.
`showSubTotals`	boolean	`true` displays the calculated subtotals for sections of the report.
`sortColumn`	string	Specifies the field on which to sort data in the report. Use `sortOrder` to specify the sort order.
`sortOrder`	SortOrder (enumeration of type string)	Specifies the sort order. Use `sortColumn` to specify the field on which to sort.

Field	Field Type	Description
territoryHierarchyFilter	string	The territory name for a report drill down. If your organization uses territory management, some reports display Hierarchy links that allow you to drill down to different datasets based on the territory hierarchy. This field is available in API version 17.0 and later.
timeFrameFilter	ReportTimeFrameFilter	Limits report results to records within a specified time frame.
userFilter	string	The username for a report drill down. Some reports, such as opportunity and activity reports, display Hierarchy links that allow you to drill down to different datasets based on the user hierarchy. This field is available in API version 17.0 and later.

ReportAggregateFilter

ReportAggregateFilter defines custom summary formula filters on summary, matrix, and joined reports.

Field	Field Type	Description
aggregate	string	Required. The name of the report aggregate to apply the filter to.
operator	string	Required. The filter operator.
value	string	Required. The filter value.

ReportAggregate

ReportAggregate defines custom summary formulas on summary, matrix, and joined reports. For more information on these fields, see Add a Summary Formula Column to a Report in Salesforce Help.

Field	Field Type	Description
acrossGroupingContext	string	Defines the row grouping level at which you want your custom summary formula to be displayed. This field is available in API version 15.0.
calculatedFormula	string	Required. The custom summary formula. For example, `AMOUNT:SUM + OPP_QUANTITY:SUM`
datatype	ReportAggregateDatatype (enumeration of type string)	Required. Specifies the data type for formatting and display of the custom summary formula results.

Field	Field Type	Description
`description`	string	The custom summary formula description. Maximum: 255 characters.
`developerName`	string	Required. The internal development name of the custom summary formula, for example, `FORMULA1`. This name is used to reference custom summary formulas from other report components, including conditional highlighting.
`downGroupingContext`	string	Defines the column grouping level at which you want your custom summary formula to be displayed. This field is available in API version 15.0 and later.
`isActive`	boolean	Required. `true` displays the formula result in the report. `false` doesn't display the result in the report.
`isCrossBlock`	boolean	Determines whether the custom summary formula is a cross-block formula, which is available with joined reports. `true` indicates a cross-block custom summary formula. `false` indicates a standard custom summary formula. This field is available in API version 25.0 and later.
`masterLabel`	string	Required. The custom summary formula label (name).
`reportType`	string	Required for joined reports. Specifies the `reportType` of the blocks to which the `aggregate` can be added.
`scale`	int	The formula result is calculated to the specified number of decimal places. Valid values `0` through `18`.

ReportBlockInfo

ReportBlockInfo defines blocks in a joined report.

Field	Field Type	Description
`aggregateReferences`	ReportAggregateReference[]	Lists the `aggregates` that represent the custom summary formulas used in a joined report block.
`blockId`	string	Required. `blockId` is used in cross-block custom summary formulas and joined report charts to identify the block containing each summary field. `blockId` is assigned automatically. Valid values are B1 through B5. This field is available in API version 25.0 and later.
`joinTable`	string	Required. Refers to the entity used to join blocks in a joined report. The entity provides a list of fields that are available for globally grouping across the blocks.

ReportAggregateReference

ReportAggregateReference defines the developer name used for custom summary formulas in joined reports.

Field	Field Type	Description
`aggregate`	string	Required. The `developerName` of the ReportAggregate, which specifies the custom summary formula used in a block of a joined report.

ReportBucketField

ReportBucketField defines a bucket to be used in the report.

Field	Field Type	Description
`bucketType`	ReportBucketFieldType (enumeration of type string)	Required. Specifies the type of bucket. Valid values: • text • number • picklist
`developerName`	string	Required. A unique name used as the `<field>` value to display a bucket field in the column list and other report components, including sort, filter, list, group, and chart. Must be of the format `BucketField_`***`name`***. For example, `BucketField_BusinessSize`.
`masterLabel`	string	Required. The bucket field label. Maximum 40 characters. Any line breaks, tabs, or multiple spaces at the beginning or end of the label are removed. Any of these characters within the label are reduced to a single space.
`nullTreatment`	ReportBucketFieldNullTreatment (enumeration of type string)	For numeric bucket fields only. Specifies whether empty values are treated as zeros (`z`) or not (`n`).
`otherBucketLabel`	string	The label of the container for unbucketed values.
`sourceColumnName`	string	Required. The source field that the bucket is applied to. For example, `SALES` or `INDUSTRY`.
`values`	ReportBucketFieldValue (enumeration of type string)	Defines one bucket value used in the bucket field. While this name is plural, it represents a single bucket. In typical use, a bucket field contains multiple buckets.

ReportBucketFieldValue

ReportBucketFieldValue defines a bucket value used in the bucket field.

Field	Field Type	Description
`sourceValues`	ReportBucketFieldSourceValue (enumeration of type string)	The value of a bucket in the bucket field. Valid values: • `sourceValue`—Used for picklist and text bucket fields. For picklists, describes the picklist item in the bucket. For example, the sourceValue of a bucket on `TYPE` could be `Customer`. For text, the full string for the item in the bucket. For example, the sourceValue of a bucket on `ADDRESS_STATE1` could be `NY`. • `from`—Used only on numeric bucket fields. A non-inclusive lower bound for a numeric bucket range. This value must be a number. • `to`—Used only on numeric bucket fields. The inclusive upper bound for a numeric bucket range. This value must be a number. In numeric buckets, the first value must only have `to` and last value must only have `from`. All other values must have both `to` and `from`.
`value`	string	Required. The name of a specific bucket value within the bucket field.

ReportGrouping

ReportGrouping defines how to group, subtotal, and sort data for summary, matrix, and joined reports.

Field	Field Type	Description
`aggregateType`	ReportAggrType (enumeration of type string)	The type of aggregate value to sort by. Valid values are: • `Sum` • `Average` • `Maximum` • `Minimum` • `RowCount` • `Unique` • `Median` • `Noop`
`dateGranularity`	UserDateGranularity (enumeration of type string)	When grouping by a date field, the time period by which to group.
`field`	string	Required. The field by which you want to summarize data. For example, `CAMPAIGN_SOURCE`
`sortByName`	string	The API name of the column, aggregate, or custom summary field used to order the grouping.

Field	Field Type	Description
`sortOrder`	SortOrder	Required. Whether to sort data in ascending or descending alphabetical and numerical order.
`sortType`	ReportSortType (enumeration of type string)	Indicates if the grouping is sorted by a column, aggregate, or custom summary field. Valid values are: • `Column` • `Aggregate` • `CustomSummaryFormula`

ReportHistoricalSelector

ReportHistoricalSelector defines a date range for historical data.

Field	Field Type	Description
`snapshot`	string	Represents the date value to apply a historical filter, either relative (in the format `N_DAYS_AGO:2`) or absolute (in the format `yyyy-MM-dd`). If unspecified, it's assumed that the filter is applied to all the columns the user sees. Available in API version 29.0 and later.

CustomDetailFormulas

CustomDetailFormulas defines row-level formulas for reports.

Field	Field Type	Description
`calculatedFormula`	string	Required. The custom formula. For example, `AMOUNT:SUM + OPP_QUANTITY:SUM`
`datatype`	ReportCustomDetailFormulaDatatype (enumeration of type string)	Required. Specifies the data type for formatting and display of the formula results.
`description`	string	The formula description. Maximum: 255 characters.
`developerName`	string	Required. The internal development name of the formula, for example, `FORMULA1`. This name is used to reference custom formulas from other report components, including conditional highlighting.
`label`	string	Required. The name that identifies this formula.
`scale`	int	The formula result is calculated to the specified number of decimal places. Valid values `0` through `18`.

ReportCustomDetailFormulaDatatype

An enumeration of type string that specifies the data type for formatting and display of row-level formula results. Valid values:

Enumeration Value
`Double`
`DateOnly`
`DateTime`
`Text`

SortOrder

An enumeration of type string that defines the order in which data is sorted in the report fields. Valid values:

Field	Description
`Asc`	Sorts data in ascending alphabetical and numerical order.
`Desc`	Sorts data in descending alphabetical and numerical order.

UserDateGranularity

An enumeration of type string that defines the time period by which to group data. Valid values:

Enumeration Value	Description
`None`	No grouping by date
`Day`	By day
`Week`	By week
`Month`	By month
`Quarter`	By quarter
`Year`	By year
`FiscalQuarter`	By fiscal quarter. You can set the fiscal year for your organization. See Set the Fiscal Year in Salesforce Help.
`FiscalYear`	By fiscal year
`MonthInYear`	By calendar month in year
`DayInMonth`	By calendar day in month
`FiscalPeriod`	When custom fiscal years are enabled: By fiscal period
`FiscalWeek`	When custom fiscal years are enabled: By fiscal week

ReportSummaryType

An enumeration of type string that defines how report fields are summarized. Valid values:

Enumeration Value	Description
`Sum`	Total
`Average`	Average
`Maximum`	Largest value
`Minimum`	Smallest value
`Unique`	Unique values
`Median`	Median value
`Noop`	The summary is a no-op.
`None`	The field isn't summarized.

ReportColorRange

ReportColorRange defines conditional highlighting for report summary data.

Field	Field Type	Description
`aggregate`	ReportSummaryType (enumeration of type string)	Required. Defines how the field specified in `columnName` is summarized. For example, `Sum`.
`columnName`	string	Required. Specifies the field whose value ranges are represented by background colors.
`highBreakpoint`	double	Required. Specifies the number that separates the mid color from the high color.
`highColor`	string	Required. Specifies the color (in HTML format) to represent data that falls into the high number range. This color spans from the `highBreakpoint` value.
`lowBreakpoint`	double	Required. Specifies the number that separates the low color from the mid color.
`lowColor`	string	Required. Specifies a color (in HTML format) to represent data that falls into the low value range, below the `lowBreakpoint` value.
`midColor`	string	Required. Specifies a color (in HTML format) to represent data that falls into the mid value range.

ReportColumn

ReportColumn defines how fields (columns) are displayed in the report.

Field	Field Type	Description
aggregateTypes	ReportSummaryType[] (enumeration of type string)	List that defines if and how each report field is summarized.
field	string	Required. The field name. For example, AGE or OPPORTUNITY_NAME
reverseColors	boolean	In historical trend reports, displays greater Date values as green and greater Amount values as red, reversing the default colors. Available in API version 29.0 and later.
showChanges	boolean	In historical trend reports, adds a column displaying the difference between current and historical Date and Amount values. Available in API version 29.0 and later.

ReportFilter

ReportFilter limits the report results by filtering data on specified fields.

Field	Field Type	Description
booleanFilter	string	Specifies filter logic conditions.

criteriaItems	ReportFilterItem	The criteria by which you want to filter report data, either by comparing historical values or by applying a date range.
		`<criteriaItems>`
		`  criteriaItems ReportFilterItem`
		`<column>Opportunity.Opportunity__hd$Amount__hst</column>`
		`  <columnToColumn>false</columnToColumn>`
		`  <operator>equals</operator>`
		`  <snapshot>N_DAYS_AGO:90</snapshot>`
		`  <value>100</value>`
		`</criteriaItems>`

language	Language (enumeration of type string)	The language used when a report filters against a picklist value using the operators contains or startsWith. For a list of valid language values, see Language.

ReportFilterItem

ReportFilterItem limits the report results by filtering data on specified fields.

Field	Field Type	Description
`column`	string	Required. The field in which to filter data. For example, `AMOUNT`
`columnToColumn`	boolean	Indicates whether the filter is a column-to-column (field-to-field) filter. Available in API version 29.0 and later for historical trending reports. Available in API version 48.0 and later for general reports.
`isUnlocked`	boolean	Optional. Indicates whether the report filter is unlocked (`true`) or locked (`false`). You can edit unlocked filters on the report run page in Lightning Experience. If unspecified, the default value is `false`. Available in API version 38.0 and later.
`operator`	FilterOperation (enumeration of type string)	Required. An enumeration of type string that defines the operator used to filter the data, for example, `greaterThan`. Valid values are: • `equals` • `notEqual` • `lessThan` • `greaterThan` • `lessOrEqual` • `greaterOrEqual` • `contains` • `notContain` • `startsWith` • `includes` • `excludes` • `within` (`DISTANCE` criteria only)
`snapshot`	string	Represents the date value, either relative (in the format `N_DAYS_AGO:2`) or absolute (in the format `yyyy-MM-dd`). Available in API version 29.0 and later.
`value`	string	The value by which you want to filter the data, for example, `1000`. The Metadata API filter condition values don't always match the values that you enter in the report wizard. For example, in the Metadata API dates are always converted to the US date format and values entered in a non-US English language can be converted to a standard US English equivalent.

ReportFormat

An enumeration of type string that defines the report format. Valid values:

Enumeration Value	Description
`Matrix`	Summarizes data in a grid. Use to compare related totals.
`Summary`	Lists, sorts, and subtotals data.
`Tabular`	Lists data with no sorting or subtotals.
`Joined`	Joins data from different report types storing each report's data in its own block.

ReportFormattingRule

Defines conditional highlighting for report summary data. You can specify up to 5 formatting rules per report.

Field	Field Type	Description
`aggregate`	ReportFormattingSummaryType (enumeration of type string)	Defines how the field specified in `columnName` is summarized. For example, `Sum`.
`columnName`	string	Required. Specifies the field whose value ranges are represented by colors.
`values`	ReportFormattingRuleValue (enumeration of type string)	Required. Specifies the background colors and associated ranges for formatted data values.

ReportFormattingSummaryType

An enumeration of type string that defines how report fields are summarized. Valid values:

Enumeration Value	Description
`Sum`	Total
`Average`	Average
`Maximum`	Largest value
`Minimum`	Smallest value
`Unique`	Unique values

ReportFormattingRuleValue

Specifies the background colors and associated ranges for formatted data values. You can specify up to 3 background colors and 0–3 range upper bounds. Valid values:

Field	Field Type	Description
`backgroundColor`	string	(Required) Specifies a highlighting color for the field in `columnName`. Must be a valid hex color string such as #54C254. At least one color is required. You can optionally specify

Field	Field Type	Description
		a different color for up to 3 ranges as determined by `rangeUpperBound`. If you don't specify a color for a particular range, the background is transparent.

`rangeUpperBound`	double	Delineates a range to which a background color applies. If you don't specify an upper bound for a particular range, the bound is assumed to be plus infinity. The following example sets the background color for the Sales column to #B50E03 for aggregate sales less than or equal to 100, sets no background for sales from 100 to 1000, and sets the background color to #006714 for sales greater than 1000.
		`<formattingRules>`
		`<aggregate>Sum</aggregate>`
		`<columnName>Sales</columnName>`
		`<values>`
		`<backgroundColor>#B50E03</backgroundColor>`
		`<rangeUpperBound>100.0</rangeUpperBound>`
		`</values>`
		`<values>`
		`<rangeUpperBound>1000.0</rangeUpperBound>`
		`</values>`
		`<values>`
		`<backgroundColor>#006714</backgroundColor>`
		`</values>`
		`</formattingRules>`

ReportParam

ReportParam represents settings specific to a report type, especially options that let you filter a report to certain useful subsets.

Field	Field Type	Description
`name`	string	Required. Specifies a specific `reportType` setting.
`value`	string	Required. The setting value.

ReportAggregateDatatype

An enumeration of type string that specifies the data type for formatting and display of custom summary formula results. Valid values:

Enumeration Value
currency
number
percent

ReportChart

ReportChart represents charts on summary, matrix, and joined reports.

Field	Field Type	Description
backgroundColor1	string	Specifies the beginning color (in HTML format) for a gradient color background.
backgroundColor2	string	Specifies the end color (in HTML format) for a gradient color background.
backgroundFadeDir	ChartBackgroundDirection (enumeration of type string)	Specifies the direction for a gradient color background. Use with `backgroundColor1` to specify the beginning color and `backgroundColor2` to specify the end color for the gradient design. Use white for both if you don't want a background design. Valid values: • `Diagonal` • `LeftToRight` • `TopToBottom`
chartSummaries	ChartSummary[]	Specifies the summaries you want to use for the chart. Invalid summaries are ignored without notification. If there are no valid summaries, RowCount is used by default for the axis value. This field is available in API version 17.0 and later.
chartType	ChartType (enumeration of type string)	Required. Specifies the chart type. Available chart types depend on the `report type`.
enableHoverLabels	boolean	Specifies whether to display values, labels, and percentages when hovering over charts. Hover details depend on chart type. Percentages apply to pie, donut, and funnel charts only. This field is available in API version 17.0 and later.
expandOthers	boolean	Specifies whether to combine all groups less than or equal to 3% of the total into a single 'Others' wedge or segment. Only applies to pie, donut, and funnel charts. Set to `true` to show all values individually on the chart; set to `false` to combine small groups into 'Others.' This field is available in API version 17.0 and later.

Field	Field Type	Description
groupingColumn	string	Specifies the field by which to group data. This data is displayed on the X-axis for vertical column charts and on the Y-axis for horizontal bar charts.
legendPosition	ChartLegendPosition (enumeration of type string)	Required. The location of the legend with respect to the chart. The valid values are: • Bottom • OnChart • Right
location	ChartPosition (enumeration of type string)	Required. Specifies whether the chart is displayed at the top or bottom of the report.
secondaryGroupingColumn	string	For grouped chart types: Specifies the field by which to group the data.
showAxisLabels	boolean	For bar and line charts: Specifies whether the chart displays names for each axis.
showPercentage	boolean	Indicates if percentages are displayed for wedges and segments of pie, donut, and funnel charts, as well as for gauges (true), or not (false).
showTotal	boolean	Indicates if the total is displayed for donut charts and gauges (true), or not (false).
showValues	boolean	Indicates if the values of individual records or groups are displayed for charts (true), or not (false).
size	ReportChartSize (enumeration of type string)	Required. Specifies the chart size.
summaryAggregate	ReportSummaryType (enumeration of type string)	Defines how to summarize the chart data. For example, Sum. No longer supported in version API 17.0 and later. See chartSummaries.
summaryAxisManualRangeEnd	double	When specifying the axis range manually: Defines the ending value.
summaryAxisManualRangeStart	double	When specifying the axis range manually: Defines the starting value.
summaryAxisRange	ChartRangeType (enumeration of type string)	Required. For bar, line, and column charts: Defines whether to specify the axis range manually or automatically.
summaryColumn	string	Required. Specifies the field by which to summarize the chart data. Typically this field is displayed on the Y-axis. No longer supported in version API 17.0 and later. See chartSummaries.
textColor	string	The color (in HTML format) of the chart text and labels.

Field	Field Type	Description
textSize	int	The size of the chart text and labels. Valid values: • 8 • 9 • 10 • 12 • 14 • 18 • 24 • 36 The maximum size is 18. Larger values are shown at 18 points.
title	string	The chart title. Max 255 characters.
titleColor	string	The color (in HTML format) of the title text.
titleSize	int	The size of the title text. Valid values: • 8 • 9 • 10 • 12 • 14 • 18 • 24 • 36 The maximum size is 18. Larger values are shown at 18 points.

ChartType

An enumeration of type string that defines the chart type. For information on each of these chart types, see Chart Types in Salesforce Help. Valid values:

Enumeration Value
None
HorizontalBar
HorizontalBarGrouped
HorizontalBarStacked
HorizontalBarStackedTo100
VerticalColumn

Enumeration Value
VerticalColumnGrouped
VerticalColumnStacked
VerticalColumnStackedTo100
Line
LineGrouped
LineCumulative
LineCumulativeGrouped
Pie
Donut
Funnel
Scatter
ScatterGrouped
VerticalColumnLine
VerticalColumnGroupedLine
VerticalColumnStackedLine
Plugin Reserved for future use. This value is available in API version 31.0 and later.

ChartPosition

An enumeration of type string that specifies the position of the chart in the report. Valid values:

Enumeration Value
CHART_TOP
CHART_BOTTOM

ChartSummary

ChartSummary defines how data in the chart is summarized. Valid values:

Field	Field Type	Description
aggregate	ReportSummaryType	Specifies the aggregation method—such as Sum, Average, Min, and Max—for the summary value. Use the column field to specify the summary value to use for the aggregation.

Field	Field Type	Description
		You don't need to specify this field for RowCount or custom summary formulas.
`axisBinding`	ChartAxis	Specifies the axis or axes to use on the chart. Use the `column` field to specify the summary value to use for the axis.
`column`	string	Required. Specifies the summary field for the chart data. If all columns are invalid, RowCount is used by default for the axis value. For vertical column and horizontal bar combination charts, you can specify up to four values.

ChartAxis

An enumeration of type string that specifies the axis or axes to be used in charts. Valid values:

Enumeration Value	Description
`x`	The summary value to use for the X-axis of a scatter chart.
`y`	The Y-axis for the chart.
`y2`	The secondary Y-axis for vertical column combination charts with a line added.

ReportChartSize

An enumeration of type string that specifies the chart size. Valid values:

Enumeration Value
`Tiny`
`Small`
`Medium`
`Large`
`Huge`

ChartRangeType

An enumeration of type string that defines the report format. Valid values:

Enumeration Value
`Auto`
`Manual`

ReportTimeFrameFilter

ReportTimeFrameFilter represents the report time period.

Field	Field Type	Description
`dateColumn`	string	Required. The date field on which to filter data. For example, `CLOSE_DATE`
`endDate`	date	When `interval` is `INTERVAL_CUSTOM`, specifies the end of the custom time period.
`interval`	UserDateInterval (enumeration of type string)	Required. Specifies the period.
`startDate`	date	When `interval` is `INTERVAL_CUSTOM`, specifies the start of the custom time period.

ReportCrossFilter

ReportCrossFilter represents the cross filter functionality in reports.

Field	Field Type	Description
`criteriaItems`	ReportFilterItem	Represents the subfilters of a cross filter. There can be up to five subfilters. This field requires the following attributes. • `Column` • `Operator` • `Value`
`operation`	ObjectFilterOperator (Enumeration of type string)	The action indicating whether to include or exclude an object. Valid values: `with` and `without`.
`primaryTableColumn`	string	The field from the parent object used for the cross filter.
`relatedTable`	string	The child object used for the cross filter.
`relatedTableJoinColumn`	string	The field from the child object that is used to join the parent.

Declarative Metadata Sample Definition

A sample XML snippet using cross filters to build an Accounts report for cases where case status isn't closed:

```
    <crossFilters>
        <criteriaItems>
            <column>Status</column>
            <operator>notequal</operator>
            <value>Closed</value>
        </criteriaItems>
        <operation>with</operation>
        <primaryTableColumn>ACCOUNT_ID</primaryTableColumn>
```

```
        <relatedTable>Case</relatedTable>
        <relatedTableJoinColumn>Account</relatedTableJoinColumn>
    </crossFilters>
```

Note: This sample was generated using the API version 23.0.

UserDateInterval

An enumeration of type string that defines the period. Valid values:

Enumeration Value	Description
INTERVAL_CURRENT	Current fiscal quarter
INTERVAL_CURNEXT1	Current and next fiscal quarters
INTERVAL_CURPREV1	Current and previous fiscal quarters
INTERVAL_NEXT1	Next fiscal quarter
INTERVAL_PREV1	Previous fiscal quarter
INTERVAL_CURNEXT3	Current and next three fiscal quarters
INTERVAL_CURFY	Current fiscal year
INTERVAL_PREVFY	Previous fiscal year
INTERVAL_PREV2FY	Previous two fiscal years
INTERVAL_AGO2FY	Two fiscal years ago
INTERVAL_NEXTFY	Next fiscal year
INTERVAL_PREVCURFY	Current and previous fiscal years
INTERVAL_PREVCUR2FY	Current and previous two fiscal years
INTERVAL_CURNEXTFY	Current and next fiscal year
INTERVAL_CUSTOM	A custom time period. Use `startDate` and `endDate` fields to specify the time period's start date and end date.
INTERVAL_YESTERDAY	Yesterday
INTERVAL_TODAY	Today
INTERVAL_TOMORROW	Tomorrow
INTERVAL_LASTWEEK	Last calendar week
INTERVAL_THISWEEK	This calendar week
INTERVAL_NEXTWEEK	Next calendar week
INTERVAL_LASTMONTH	Last calendar month
INTERVAL_THISMONTH	This calendar month

Enumeration Value	Description
INTERVAL_NEXTMONTH	Next calendar month
INTERVAL_LASTTHISMONTH	Current and previous calendar months
INTERVAL_THISNEXTMONTH	Current and next calendar months
INTERVAL_CURRENTQ	Current calendar quarter
INTERVAL_CURNEXTQ	Current and next calendar quarters
INTERVAL_CURPREVQ	Current and previous calendar quarters
INTERVAL_NEXTQ	Next calendar quarter
INTERVAL_PREVQ	Previous calendar quarter
INTERVAL_CURNEXT3Q	Current and next three calendar quarters
INTERVAL_CURY	Current calendar year
INTERVAL_PREVY	Previous calendar year
INTERVAL_PREV2Y	Previous two calendar years
INTERVAL_AGO2Y	Two calendar years ago
INTERVAL_NEXTY	Next calendar year
INTERVAL_PREVCURY	Current and previous calendar years
INTERVAL_PREVCUR2Y	Current and previous two calendar years
INTERVAL_CURNEXTY	Current and next calendar years
INTERVAL_LAST7	Last 7 days
INTERVAL_LAST30	Last 30 days
INTERVAL_LAST60	Last 60 days
INTERVAL_LAST90	Last 90 days
INTERVAL_LAST120	Last 120 days
INTERVAL_NEXT7	Next 7 days
INTERVAL_NEXT30	Next 30 days
INTERVAL_NEXT60	Next 60 days
INTERVAL_NEXT90	Next 90 days
INTERVAL_NEXT120	Next 120 days
LAST_FISCALWEEK	When custom fiscal years are enabled: Last fiscal week
THIS_FISCALWEEK	When custom fiscal years are enabled: This fiscal week
NEXT_FISCALWEEK	When custom fiscal years are enabled: Next fiscal week

Enumeration Value	Description
LAST_FISCALPERIOD	When custom fiscal years are enabled: Last fiscal period
THIS_FISCALPERIOD	When custom fiscal years are enabled: This fiscal period
NEXT_FISCALPERIOD	When custom fiscal years are enabled: Next fiscal period
LASTTHIS_FISCALPERIOD	When custom fiscal years are enabled: This fiscal period and last fiscal period
THISNEXT_FISCALPERIOD	When custom fiscal years are enabled: This fiscal period and next fiscal period
CURRENT_ENTITLEMENT_PERIOD	Current entitlement period
PREVIOUS_ENTITLEMENT_PERIOD	Previous entitlement period
PREVIOUS_TWO_ENTITLEMENT_PERIODS	Previous two entitlement periods
TWO_ENTITLEMENT_PERIODS_AGO	Two entitlement periods ago
CURRENT_AND_PREVIOUS_ENTITLEMENT_PERIOD	Current and previous entitlement period
CURRENT_AND_PREVIOUS_TWO_ENTITLEMENT_PERIODS	Current and previous two entitlement periods

Declarative Metadata Sample Definition

A sample XML report definition:

```
<?xml version="1.0" encoding="UTF-8"?>
<Report xmlns="http://soap.sforce.com/2006/04/metadata">
    <aggregates>
        <acrossGroupingContext>CRT_Object__c$Id</acrossGroupingContext>
        <calculatedFormula>PREVGROUPVAL(CRT_Object__c.Currency__c:AVG, CRT_Object__c.Id)
*
                PARENTGROUPVAL(CRT_Object__c.Number__c:MAX, CRT_Object__c.CreatedBy.Name,

                COLUMN_GRAND_SUMMARY)/RowCount</calculatedFormula>
        <datatype>number</datatype>
        <developerName>FORMULA1</developerName>
        <downGroupingContext>CRT_Object__c$CreatedBy</downGroupingContext>
        <isActive>true</isActive>
        <masterLabel>CurrCSF</masterLabel>
        <scale>2</scale>
    </aggregates>
    <aggregates>
        <acrossGroupingContext>CRT_Object__c$LastModifiedDate</acrossGroupingContext>
        <calculatedFormula>IF(RowCount&gt;10,
                BLANKVALUE(ROUND(PREVGROUPVAL(CRT_Object__c.Currency__c:SUM,
                CRT_Object__c.LastModifiedDate),3),
                PARENTGROUPVAL(CRT_Object__c.Number__c:SUM, ROW_GRAND_SUMMARY,
                CRT_Object__c.Id))  , 1000)</calculatedFormula>
        <datatype>number</datatype>
        <developerName>FORMULA2</developerName>
        <downGroupingContext>GRAND_SUMMARY</downGroupingContext>
        <isActive>true</isActive>
        <masterLabel>numCSF</masterLabel>
```

```
        <scale>2</scale>
    </aggregates>
    <buckets>
        <bucketType>number</bucketType>
        <developerName>BucketField_BusinessSize</developerName>
        <masterLabel>NumericBucket</masterLabel>
        <nullTreatment>z</nullTreatment>
        <sourceColumnName>SALES</sourceColumnName>
        <values>
            <sourceValues>
                <to>10000</to>
            </sourceValues>
            <value>low</value>
        </values>
        <values>
            <sourceValues>
                <from>10000</from>
                <to>25000</to>
            </sourceValues>
            <value>mid</value>
        </values>
        <values>
            <sourceValues>
                <from>25000</from>
            </sourceValues>
            <value>high</value>
        </values>
    </buckets>
    <buckets>
        <bucketType>text</bucketType>
        <developerName>BucketField_Region</developerName>
        <masterLabel>TextBucket</masterLabel>
        <nullTreatment>n</nullTreatment>
        <otherBucketLabel>Other</otherBucketLabel>
        <sourceColumnName>ADDRESS1_STATE</sourceColumnName>
        <values>
            <sourceValues>
                <sourceValue>CA</sourceValue>
            </sourceValues>
            <value>west</value>
        </values>
        <values>
            <sourceValues>
                <sourceValue>NY</sourceValue>
            </sourceValues>
            <sourceValues>
                <sourceValue>Ontario</sourceValue>
            </sourceValues>
            <value>east</value>
        </values>
    </buckets>
    <chart>
        <backgroundColor1>#FFFFFF</backgroundColor1>
        <backgroundColor2>#FFFFFF</backgroundColor2>
```

```
        <backgroundFadeDir>Diagonal</backgroundFadeDir>
        <chartSummaries>
            <axisBinding>y</axisBinding>
            <column>FORMULA1</column>
        </chartSummaries>
        <chartSummaries>
            <axisBinding>y</axisBinding>
            <column>FORMULA2</column>
        </chartSummaries>
        <chartSummaries>
            <aggregate>Maximum</aggregate>
            <axisBinding>y</axisBinding>
            <column>CRT_Object__c$Number__c</column>
        </chartSummaries>
        <chartSummaries>
            <axisBinding>y</axisBinding>
            <column>RowCount</column>
        </chartSummaries>
        <chartType>VerticalColumn</chartType>
        <groupingColumn>CRT_Object__c$LastModifiedDate</groupingColumn>
        <legendPosition>Right</legendPosition>
        <location>CHART_TOP</location>
        <size>Medium</size>
        <summaryAxisRange>Auto</summaryAxisRange>
        <textColor>#000000</textColor>
        <textSize>12</textSize>
        <titleColor>#000000</titleColor>
        <titleSize>18</titleSize>
    </chart>
    <columns>
        <field>CRT_Object__c$Name</field>
    </columns>
    <columns>
        <aggregateTypes>Average</aggregateTypes>
        <field>CRT_Object__c$Currency__c</field>
    </columns>
    <columns>
        <aggregateTypes>Maximum</aggregateTypes>
        <field>CRT_Object__c$Number__c</field>
    </columns>
    <columns>
        <field>BucketField__Region</field>
    </columns>
    <format>Matrix</format>
    <groupingsAcross>
        <dateGranularity>Day</dateGranularity>
        <field>CRT_Object__c$Id</field>
        <sortOrder>Asc</sortOrder>
    </groupingsAcross>
    <groupingsAcross>
        <dateGranularity>Year</dateGranularity>
        <field>CRT_Object__c$LastModifiedDate</field>
        <sortOrder>Asc</sortOrder>
    </groupingsAcross>
```

```
    <groupingsDown>
        <dateGranularity>Day</dateGranularity>
        <field>CRT_Object__c$CreatedBy</field>
        <sortOrder>Asc</sortOrder>
    </groupingsDown>
    <groupingsDown>
        <dateGranularity>Day</dateGranularity>
        <field>CRT_Object__c$Currency__c</field>
        <sortOrder>Desc</sortOrder>
    </groupingsDown>
    <name>CrtMMVC</name>
    <reportType>CRT1__c</reportType>
    <scope>organization</scope>
    <showDetails>false</showDetails>
    <timeFrameFilter>
        <dateColumn>CRT_Object__c$CreatedDate</dateColumn>
        <interval>INTERVAL_CUSTOM</interval>
    </timeFrameFilter>
</Report>
```

Declarative Metadata Sample Definition for a Joined Report

A sample XML report definition:

```
<?xml version="1.0" encoding="UTF-8"?>
<Report xmlns="http://soap.sforce.com/2006/04/metadata">
<!-- This is a cross-block custom summary formula. Note that the calculated formula reference
 for a blocks reference uses the BlockId#Aggregate. -->
    <aggregates>
        <calculatedFormula>B1#AMOUNT:SUM+B2#EMPLOYEES:SUM</calculatedFormula>
        <datatype>number</datatype>
        <developerName>FORMULA</developerName>
        <isActive>true</isActive>
        <isCrossBlock>true</isCrossBlock>
        <masterLabel>Cross-Block CSF Example</masterLabel>
        <scale>2</scale>
    </aggregates>
<!-- This is a standard custom summary formula. Note that the calculated formula reference
 does not have block reference but just the aggregate name of the report type associated
(Opportunity).-->
    <aggregates>
        <calculatedFormula>AMOUNT:SUM</calculatedFormula>
        <developerName>FORMULA2</developerName>
        <isActive>true</isActive>
        <isCrossBlock>false</isCrossBlock>
        <masterLabel>Standard CSF Example</masterLabel>
        <reportType>Opportunity</reportType>
        <scale>2</scale>
    </aggregates>
    <block>
      <blockInfo>
<!-- This is how the block defines that the custom summary formula should be referenced.
In this example, it's the in standard FORMULA 2 defined above. This block report has blockID
 B1.-->
```

```
        <aggregateReferences>
          <aggregate>FORMULA2</aggregate>
        </aggregateReference>
        <blockId>B1</blockId>
        <joinTable>a</joinTable>
      </blockInfo>
      <columns>
        <field>TYPE</field>
      </columns>
      <format>Summary</format>
      <name>Opportunities BLock 3</name>
      <params>
        <name>role_territory</name>
        <value>role</value>
      </params>
      <params>
        <name>terr</name>
        <value>all</value>
      </params>
      <params>
        <name>open</name>
        <value>all</value>
      </params>
      <params>
        <name>probability</name>
        <value>0</value>
      </params>
      <params>
        <name>co</name>
        <value>1</value>
      </params>
      <reportType>Opportunity</reportType>
      <scope>organization</scope>
      <timeFrameFilter>
        <dateColumn>CLOSE_DATE</dateColumn>
        <interval>INTERVAL_CUSTOM</interval>
      </timeFrameFilter>
    </block>
    <block>
      <blockInfo>
<!-- This is how the block defines that the custom summary formula should be referenced.
In this example, it's the cross-block custom summary formula FORMULA 1 defined above. This
 block report has blockId B2.-->
        <aggregateReferences>
          <aggregate>FORMULA1</aggregate>
        </aggregateReferences>
        <blockId>B2</blockId>
        <joinTable>a</joinTable>
      </blockInfo>
      <columns>
        <field>USERS.NAME</field>
      </columns>
      <columns>
        <field>TYPE</field>
```

```
      </columns>
      <columns>
         <field>DUE_DATE</field>
      </columns>
      <columns>
        <field>LAST_UPDATE</field>
      </columns>
      <columns>
        <field>ADDRESS1_STATE</field>
      </columns>
      <format>Summary</format
      <name>Accounts block 5</name>
      <params>
        <name>terr</name>
        <value>all</value>
      </params>
      <params>
        <name>co</name>
        <value>1</value>
      </params>
      <reportType>AccountList</reportType>
      <scope>organization</scope>
      <timeFrameFilter>
        <dateColumn>CREATED_DATE</dateColumn>
        <interval>INTERVAL_CUSTOM</interval>
      </timeFrameFilter>
    </block>
    <blockInfo>
      <blockId xsi:nil="true"/>
      <joinTable>a</joinTable>
    </blockInfo>
<chart>
        <backgroundColor1>#FFFFFF</backgroundColor1>
        <backgroundColor2>#FFFFFF</backgroundColor2>
        <backgroundFadeDir>Diagonal</backgroundFadeDir>
        <chartSummaries>
            <axisBinding>y</axisBinding>
<!-- This is how chart aggregates are designed in multiblock. We're using RowCount from
Block 1.-->
            <column>B1#RowCount</column>
        </chartSummaries>
        <chartType>HorizontalBar</chartType>
        <enableHoverLabels>false</enableHoverLabels>
        <expandOthers>true</expandOthers>
        <groupingColumn>ACCOUNT_NAME</groupingColumn>
        <location>CHART_TOP</location>
        <showAxisLabels>true</showAxisLabels>
        <showPercentage>false</showPercentage>
        <showTotal>false</showTotal>
        <showValues>false</showValues>
        <size>Medium</size>
        <summaryAxisRange>Auto</summaryAxisRange>
        <textColor>#000000</textColor>
        <textSize>12</textSize>
```

```
        <titleColor>#000000</titleColor>
        <titleSize>18</titleSize>
    </chart>
    <format>MultiBlock</format>
    <groupingsDown>
        <dateGranularity>Day</dateGranularity>
        <field>ACCOUNT_NAME</field>
        <sortOrder>Asc</sortOrder>
    </groupingsDown>
    <name>mb_mbapi</name>
    <reportType>Opportunity</reportType>
    <showDetails>true</showDetails>
</Report>
```

Wildcard Support in the Manifest File

This metadata type doesn't support the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SEE ALSO:

Dashboard

ReportType

Represents the metadata associated with a custom report type. Custom report types allow you to build a framework from which users can create and customize reports.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

This type extends the Metadata metadata type and inherits its `fullName` field.

Declarative Metadata File Suffix and Directory Location

The file suffix is `.reportType` for the custom report type definition. There's one file per custom report type. Report types are stored in the `reportTypes` directory of the corresponding package directory.

Version

Custom report types are available in API version 14.0 and later.

Fields

Field Name	Field Type	Description
`autogenerated`	boolean	Indicates that the report type was automatically generated when historical trending was enabled for an entity. Available in API version 29 and later.
`baseObject`	string	Required. The primary object for the custom report type, for example, Account. All objects, including custom and external objects, are supported. You can't edit this field after initial creation. Support for external objects is available in API version 38.0 and later.
`category`	ReportTypeCategory (enumeration of type string)	This field controls the category for the report. The valid values are: • `accounts` • `opportunities` • `forecasts` • `cases` • `leads` • `campaigns` • `activities` • `busop` • `products` • `admin` • `territory` • `territory2` (This value is available in API version 31.0 and later.) • `usage_entitlement` • `wdc` (This value is available in API version 29.0 and later.) • `calibration` (This value is available in API version 29.0 and later.) • `other` • `content` • `quotes` • `individual` (This value is available in API version 45.0 and later.) • `employee` (This value is available in API version 46.0 and later.) • `data_cloud` (This value is available in API version 55.0 and later.) • `commerce` (This value is available in API version 60.0 and later.) • `flow` (This value is available in API version 60.0 and later.) • `semantic_model` (This value is available in API version 60.0 and later.)
`deployed`	boolean	Required. Indicates whether the report type is available to users (`true`) or whether it's still in development (`false`).

Field Name	Field Type	Description
`description`	string	The description of the custom report type.
`fullName`	string	The report type developer name used as a unique identifier for API access. The `fullName` can contain only underscores and alphanumeric characters. It must be unique, begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
`join`	ObjectRelationship	The object joined to the `baseObject`. For example, Contacts can be joined to the primary Accounts object.
`label`	string	Required. The report type label.
`sections`	ReportLayoutSection[]	The groups of columns available for the report type. Though columns aren't strictly required, a report without columns isn't useful.

ObjectRelationship

ObjectRelationship represents a join to another object.

Field Name	Field Type	Description
`join`	ObjectRelationship	This field is a recursive reference that allows you to join more than two objects. A maximum of four objects can be joined in a custom report type. When more than two objects are joined, an inner join isn't allowed if there has been an outer join earlier in the join sequence. The `baseObject` is first joined to the object specified in `relationship`; the resulting dataset is then joined with any objects specified in this field.
`outerJoin`	boolean	Required. Indicates whether it's an outer join (`true`) or not (`false`). An outer join returns a row even if the joined table doesn't contain a matching value in the join column.
`relationship`	string	Required. The object joined to the primary object; for example, Contacts.

ReportLayoutSection

ReportLayoutSection represents a group of columns used in the custom report type.

Field Name	Field Type	Description
`columns`	ReportTypeColumn[]	The list of columns projected from the query, defined by this custom report type.
`masterLabel`	string	Required. The label for this group of columns in the report wizard.

ReportTypeColumn

ReportTypeColumn represents a column in the custom report type.

Field Name	Field Type	Description
`checkedByDefault`	boolean	Required. Indicates whether this column is selected by default (`true`) or not (`false`).
`displayNameOverride`	string	A customized column name, if desired.
`field`	string	Required. The field name associated with the report column.
`table`	string	Required. The table associated with the field; for example, Account.

Declarative Metadata Sample Definition

The definition of a custom report type is shown in this example. Account is joined to Contacts and the resulting dataset is joined with Assets.

```
<?xml version="1.0" encoding="UTF-8"?>
<ReportType xmlns="http://soap.sforce.com/2006/04/metadata">
    <baseObject>Account</baseObject>
    <category>accounts</category>
    <deployed>true</deployed>
    <description>Account linked to Contacts and Assets</description>
    <join>
        <join>
            <outerJoin>false</outerJoin>
            <relationship>Assets</relationship>
        </join>
        <outerJoin>false</outerJoin>
        <relationship>Contacts</relationship>
    </join>
    <label>Account Contacts and Assets</label>
    <sections>
        <columns>
            <checkedByDefault>true</checkedByDefault>
            <field>obj_lookup__c.Id</field>
            <table>Account</table>
        </columns>
        <columns>
            <checkedByDefault>false</checkedByDefault>
            <field>obj_lookup__c.Name</field>
            <table>Account</table>
        </columns>
        <columns>
            <checkedByDefault>false</checkedByDefault>
            <field>Opportunity__c.Amount</field>
            <table>Account</table>
        </columns>
        <columns>
            <checkedByDefault>false</checkedByDefault>
            <field>Owner.IsActive</field>
```

```
            <table>Account</table>
        </columns>
        <masterLabel>Accounts</masterLabel>
    </sections>
    <sections>
        <columns>
            <checkedByDefault>false</checkedByDefault>
            <field>Owner.Email</field>
            <table>Account.Contacts</table>
        </columns>
        <columns>
            <checkedByDefault>false</checkedByDefault>
            <field>byr__c</field>
            <table>Account.Contacts</table>
        </columns>
        <columns>
            <checkedByDefault>true</checkedByDefault>
            <field>ReportsTo.CreatedBy.Contact.Owner.MobilePhone</field>
            <table>Account.Contacts</table>
        </columns>
        <masterLabel>Contacts</masterLabel>
    </sections>
</ReportType>
```

Usage

The custom report type refers to fields by using their API names. For a historical field (one that has `trackTrending` set to `true`) the API name includes `hst`, such as `Field2__c_hst`.

```
<sections>
    <columns>
        <checkedByDefault>false</checkedByDefault>
        <field>Field2__c_hst</field>
        <table>CustomTrendedObject__c.CustomTrendedObject__c_hst</table>
    </columns>
    <masterLabel>History</masterLabel>
</sections>
```

For more information, see `trackTrending` on page 694.

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

RestrictionRule

Represents a restriction rule or a scoping rule. A restriction rule has `enforcementType` set to `Restrict` and controls the access that specified users have to designated records. A scoping rule has `enforcementType` set to `Scoping` and controls the default records that your users see without restricting access. This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

RestrictionRule components have the suffix `.rule` and are stored in the `restrictionRules` folder.

Version

RestrictionRule components are available in API version 52.0 and later.

Special Access Rules

Only users with the View Restriction and Scoping Rules permission can view restriction rules and scoping rules via the API. Only users with the Manage Sharing permission can view, create, update, and delete restriction rules and scoping rules.

Fields

Field Name	Field Type	Description
`active`	boolean	Indicates whether the rule is active (`true`) or not (`false`). The default value is `false`.
`description`	string	Required. The description of the rule.
`enforcementType`	EnforcementType (enumeration of type string)	Required. The type of rule. Valid values are: • `FieldRestrict`—Don't use. • `Restrict`—Restriction rule. • `Scoping`—Scoping rule.
`masterLabel`	string	Required. The name of the rule.
`recordFilter`	string	Required. The criteria that determine which records are accessible via the rule. For picklist fields, you can now use the OR operator to specify multiple values for a single picklist field.— For example: OR(ISPICKVAL(Status,'Draft'), ISPICKVAL(Status,'Activated'), ISPICKVAL(Status,'Negotiating')). This enhancement applies to both Restriction Rules and Scoping Rules. Support for multiple picklist values using the OR operator was introduced in API version 60.0 and later.
`targetEntity`	string	Required. The object for which you're creating the rule. We recommend that you don't edit this field after the rule is created. If `enforcementType` is set to `Restrict`, custom objects, external objects, and these objects are supported: • Contract

Field Name	Field Type	Description
		• Event • Task • TimeSheet • TimeSheetEntry If `enforcementType` is set to `Scoping`, custom objects and these objects are supported: • Account • Case • Contact • Event • Lead • Opportunity • Task
`userCriteria`	string	Required. The users that this rule applies to, such as all active users or users with a specified role or profile.
`version`	int	Required. The rule's version number.

Declarative Metadata Sample Definition

The following is an example of a RestrictionRule component representing a restriction rule.

```
<?xml version="1.0" encoding="UTF-8"?>
<RestrictionRule xmlns="http://soap.sforce.com/2006/04/metadata">
    <active>true</active>
    <description>Allows users with a specific profile to see only tasks that they
own.</description>
    <enforcementType>Restrict</enforcementType>
    <masterLabel>Tasks You Own</masterLabel>
    <recordFilter>OwnerId = $User.Id</recordFilter>
    <targetEntity>Task</targetEntity>
    <userCriteria>$User.ProfileId = '00exxxxxxxxxxx'</userCriteria>
    <version>1</version>
</RestrictionRule>
```

The following is an example of a RestrictionRule component representing a scoping rule.

```
<?xml version="1.0" encoding="UTF-8"?>
<RestrictionRule xmlns="http://soap.sforce.com/2006/04/metadata">
    <active>true</active>
    <description>View tasks contacts from Department A.</description>
    <enforcementType>Scoping</enforcementType>
    <masterLabel>SR for Department A contacts</masterLabel>
    <recordFilter>Department=$User.Department</recordFilter>
    <targetEntity>Contact</targetEntity>
    <userCriteria>$User.UserRoleId = '00Exxxxxxxxxxx'</userCriteria>
```

```
    <version>1</version>
</RestrictionRule>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>RestrictionRule</name>
    </types>
    <version>55.0</version>
</Package>
```

Role

Represents a role in your organization.

Declarative Metadata File Suffix and Directory Location

The file suffix for role components is `.role` and components are stored in the `roles` directory of the corresponding package directory.

Version

Role components are available in API version 24.0 and later.

Fields

This metadata type extends to subtype RoleOrTerritory on page 1765.

Field Name	Field Type	Description
`fullName`	string	The unique identifier for API access. The `fullName` can contain only underscores and alphanumeric characters. It must be unique, begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. This field is inherited from the Metadata component. Corresponds to **Role Name** in the user interface.
`parentRole`	string	The role above this role in the hierarchy.

Declarative Metadata Sample Definition

The following is the definition of a role.

```
<?xml version="1.0" encoding="UTF-8"?>
<Role xmlns="http://soap.sforce.com/2006/04/metadata">
    <caseAccessLevel>Edit</caseAccessLevel>
    <contactAccessLevel>Edit</contactAccessLevel>
```

```
    <description>Sample Role</description>
    <mayForecastManagerShare>false</mayForecastManagerShare>
    <name>R22</name>
    <opportunityAccessLevel>Read</opportunityAccessLevel>
</Role>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

RoleOrTerritory

Represents the common base type and valid values for role or territory.

Version

RoleOrTerritory components are available in API version 24.0 and later.

Note: You can't create a RoleOrTerritory component directly. Use the Role or Territory metadata types instead.

Fields

Field Name	Field Type	Description
`caseAccessLevel`	string	Specifies whether a user can access other users' cases that are associated with accounts the user owns. Valid values are: • `Read` • `Edit` • `None` This field is not visible if your organization's sharing model for cases is Public Read/Write. If no value is set for this field, this field value uses the default access level that is specified in the Manage Territory page in Setup.
`contactAccessLevel`	string	Specifies whether a user can access other users' contacts that are associated with accounts the user owns. Valid values are: • `Read` • `Edit` • `None` This field is not visible if your organization's sharing model for contacts is Public Read/Write or Controlled by Parent. If no value is set for this field, this field value uses the default access level that is specified in the Manage Territory page in Setup.

Field Name	Field Type	Description
`description`	string	The description of the role or territory.
`fullName`	string	The unique identifier for API access. The `fullName` can contain only underscores and alphanumeric characters. It must be unique, begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. This field is inherited from the Metadata component.
`mayForecastManagerShare`	boolean	Indicates whether the forecast manager can manually share their own forecast.
`name`	string	Required. The name of the role or territory.
`opportunityAccessLevel`	string	Specifies whether a user can access other users' opportunities that are associated with accounts the user owns. Valid values are: • `Read` • `Edit` • `None` This field is not visible if your organization's sharing model for opportunities is Public Read/Write. If no value is set for this field, this field value uses the default access level that is specified in the Manage Territory page in Setup.

Declarative Metadata Sample Definition

The following is the definition of a role.

```
<?xml version="1.0" encoding="UTF-8"?>
<Role xmlns="http://soap.sforce.com/2006/04/metadata">
    <caseAccessLevel>Edit</caseAccessLevel>
    <contactAccessLevel>Edit</contactAccessLevel>
    <description>Sample Role</description>
    <mayForecastManagerShare>false</mayForecastManagerShare>
    <name>R22</name>
    <opportunityAccessLevel>Read</opportunityAccessLevel>
</Role>
```

The following is the definition of a territory.

```
<?xml version="1.0" encoding="UTF-8"?>
<Territory xmlns="http://soap.sforce.com/2006/04/metadata">
    <accountAccessLevel>Edit</accountAccessLevel>
    <caseAccessLevel>Edit</caseAccessLevel>
    <contactAccessLevel>Edit</contactAccessLevel>
    <description>Sample Territory</description>
    <mayForecastManagerShare>false</mayForecastManagerShare>
    <name>T22name</name>
```

```
    <opportunityAccessLevel>Read</opportunityAccessLevel>
</Territory>
```

SEE ALSO:

- Role
- Territory

SalesWorkQueueSettings

Represents settings used to customize work queue options for third-party scoring. In Sales Engagement, you can add a custom number field on person accounts, contacts, or leads. Then, use the custom number field to sort the work queue. This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

SalesWorkQueueSettings components have the suffix `.salesworkqueuesetting` and are stored in the `salesworkqueuesettings` folder.

Version

SalesWorkQueueSettings components are available in API version 49.0 and later.

Special Access Rules

You must be a Sales Engagement customer to access this metadata type.

Fields

Field Name	Field Type	Description
`featureName`	string	The feature that the SalesWorkQueueSettings record is configuring. The allowed value is `ThirdPartyScore`.
`targetEntity`	string	The type that the SalesWorkQueueSettings record is configuring. Possible values are: • `Contact` • `Lead` • `PersonAccount`
`targetField`	string	The developer name or ID of the custom number field that is used to sort the work queue. Custom fields must have a custom number data type. • To use Einstein Intelligence Score for lead scoring, use `ScoreIntelligence.Score` for the developer name. • To remove custom number fields from the work queue, use `None`.

Declarative Metadata Sample Definition

The following is an example of a SalesWorkQueueSettings component. The value for `targetField` is set to `00NRM000001g55D` as an example of a custom field ID. Replace this value with the ID of your custom field.

```
<?xml version="1.0" encoding="UTF-8"?>
<SalesWorkQueueSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <featureName>ThirdPartyScore</featureName>
    <targetEntity>Contact</targetEntity>
    <targetField>00NRM000001g55D</targetField>
</SalesWorkQueueSettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>SalesWorkQueueSettings</name>
    </types>
    <version>49.0</version>
</Package>
```

Usage

Create one SalesWorkQueueSettings record for each type. For example, suppose that you want to create a work queue to sort leads by your custom field called `customLeadScore`. Create a SalesWorkQueueSettings record and set `featureName` to `ThirdPartyScore`, `targetEntity` to `Lead`, and `targetField` to `customLeadScore`.

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SamlSsoConfig

Represents a SAML Single Sign-On configuration. This type extends the Metadata metadata type and inherits its `fullName` field. Single sign-on (SSO) is an authentication method that enables users to access multiple applications with one login and one set of credentials. For example, after users log in to your org, they can automatically access all apps from the App Launcher. You can set up your Salesforce org to trust a third-party identity provider to authenticate users. Or you can configure a third-party app to rely on your org for authentication.

File Suffix and Directory Location

SamlSsoConfig components have the suffix `.samlssoconfig` and are stored in the `samlssoconfigs` folder.

Version

SamlSsoConfig components are available in API version 28.0 and later.

Special Access Rules

As of Summer '20 and later, only users with the View Setup and Configuration permission or both the Customize Application and Modify All Data permissions can access this type.

Fields

Field Name	Field Type	Description
`attributeNameIdFormat`	string	For SAML 2.0, only and when `identityLocation` is set to `Attribute`. Possible values include `unspecified`, `emailAddress`, or `persistent`. All legal values can be found in the "Name Identifier Format Identifiers" section of the Assertions and Protocols SAML 2.0 specification.
`attributeName`	string	The name of the identity provider's application. Get this name from your identity provider.
`decryptionCertificate`	string	The name of the certificate to use for decrypting incoming SAML assertions. This certificate is saved in the organization's Certificate and Key Management list. Available in API version 30.0 and later.
`errorUrl`	string	When there's an error during login, specify the URL of the page where users are directed. It must be publicly accessible, such as a public site Visualforce page. The URL can be absolute or relative.
`executionUserId`	string	The user that runs the Apex handler class. The user must have the Manage Users permission. If you specify a SAML JIT handler class, a user is required.
`identityLocation`	SamlIdentityLocationType (enumeration of type string)	The location in the assertion where a user is identified. Valid values are: • `SubjectNameId` — The identity is in the `<Subject>` statement of the assertion. • `Attribute` — The identity is specified in an `<AttributeValue>`, located in the `<Attribute>` of the assertion.
`identityMapping`	SamlIdentityType (enumeration of type string)	The identifier the service provider uses for the user during Just-in-Time user provisioning. Valid values are: • `Username` — The user's Salesforce username. • `FederationId` — The federation ID from the user object; the identifier used by the service provider for the user. • `UserId` — The user ID from the user's Salesforce organization.
`issuer`	string	The identification string for the Identity Provider.
`loginUrl`	string	For SAML 2.0 only: The URL where Salesforce sends a SAML request to start the login sequence.
`logoutUrl`	string	For SAML 2.0 only: The URL to direct the user to when they click the Logout link. The default is `https://salesforce.com`.

Field Name	Field Type	Description
`name`	string	The unique name used by the API and managed packages. The name must begin with a letter and use only alphanumeric characters and underscores. The name cannot end with an underscore or have two consecutive underscores.
`oauthTokenEndpoint`	string	For SAML 2.0 only: The ACS URL used with enabling Salesforce as an identity provider in the web single sign-on OAuth assertion flow.
`redirectBinding`	boolean	Choose the binding mechanism your identity provider requests for your SAML messages. Values are: • `HTTP POST` — HTTP POST binding sends SAML messages using base64-encoded HTML forms. • `HTTP Redirect` — HTTP Redirect binding sends base64-encoded and URL-encoded SAML messages within URL parameters.
`requestSignatureMethod`	string	The method that's used to sign the SAML request. Valid values are `RSA-SHA1` and `RSA-SHA256`.
`requestSigningCertId`	string	The 18-digit ID for the certificate used to generate the signature on a SAML request to the identity provider. The certificate is saved in the Certificate and Key Management page in Setup.
`salesforceLoginUrl`	string	The URL associated with login for the web single sign-on flow. Note: When encryption is enabled, the URL has a parameter containing the ID of the SAML configuration, `sc=`***`samlSsoConfigId`***. For example, `https://mycompany.my.salesforce.com?sc=0LEB0000000CCC`. This change applies to API Version 47.0 and later.
`samlEntityId`	string	The issuer in SAML requests generated by Salesforce, and is also the expected audience of any inbound SAML Responses. Salesforce recommends that you use your My Domain login URL.
`samlJitHandlerId`	string	The name of an existing Apex class that implements the `Auth.SamlJitHandler` interface.
`samlVersion`	SamlType (enumeration of type string)	The SAML version in use. Valid values are: • `SAML1_1` — SAML 1.1 • `SAML2_0` — SAML 2.0
`singleLogoutBinding`	SamlSpSLOBinding (enumeration of type string)	The HTTP binding type. This value determines where to put the LogoutRequest or LogoutResponse in the SAML request during single logout (SLO). The value is base64 encoded. Valid values are: • `RedirectBinding` — Sent in the query string, deflated. • `PostBinding` — Sent in the POST body, not deflated.

Field Name	Field Type	Description
singleLogoutUrl	string	The SAML single logout endpoint. This URL is the endpoint where Salesforce sends LogoutRequests (when Salesforce initiates a logout), or LogoutResponses (when the identity provider initiates a logout).
useConfigRequestMethod	boolean	If `true`, applies the selected Request Signature Method (RSM) during single logout. If `false`, the default RSM (RSA-SHA1) is applied.
useSameDigestAlgoForSigning	boolean	If `true`, uses a digest algorithm based on the selected Request Signature Method (RSM). For example, if the selected RSM is `RSA-SHA256`, the digest algorithm is set to `SHA-256`. If `false`, uses the default digest algorithm (`SHA-1`), regardless of the selected RSM. This field is available in API version 55.0 and later. You can edit this field only for legacy SAML configurations created before Spring '22. For configurations created after Spring '22, this field is `true` by default.
userProvisioning	boolean	If `true`, Just-in-Time user provisioning is enabled, which creates users the first time they log in. Specify `Federation ID` for the `identityMapping` value to use this feature.
validationCert	string	The certificate used to validate the request. Get this certificate from your identity provider.

Declarative Metadata Sample Definition

The following is an example of a SamlSsoConfig component. The validation certificate string has been truncated for readability.

```
<?xml version="1.0" encoding="UTF-8"?>
<SamlSsoConfig xmlns="http://soap.sforce.com/2006/04/metadata">
  <identityLocation>SubjectNameId</identityLocation>
  <identityMapping>FederationId</identityMapping>
  <issuer>https://my-idp.my.salesforce.com</issuer>
  <loginUrl>
    https://my-idp.my.salesforce.com/idp/endpoint/HttpRedirect
  </loginUrl>
  <logoutUrl>https://www.salesforce.com</logoutUrl>
  <name>SomeCompany</name>
  <oauthTokenEndpoint>
    https://login.salesforce.com/services/oauth2/token?so=00DD0000000
  </oauthTokenEndpoint>
  <redirectBinding>true</redirectBinding>
  <requestSignatureMethod>RSA-SHA1</requestSignatureMethod>
  <salesforceLoginUrl>
    https://login.salesforce.com?so=00DD0000000JxeI
  </salesforceLoginUrl>
  <samlEntityId>
    https://saml.salesforce.com/customPath
  </samlEntityId>
  <samlVersion>SAML2_0</samlVersion>
```

```
  <useConfigRequestMethod>true</useConfigRequestMethod>
  <userProvisioning>false</userProvisioning>
  <validationCert>
    MIIEojCCA4qgAwIBAgIOATtxsoBFAAAAAD4...
  </validationCert>
</SamlSsoConfig>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SchedulingObjective

Represents a scheduling objective in Workforce Engagement. Scheduling objectives define business goals that the scheduling tools consider when identifying agents for shifts.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

`SchedulingObjective` components have the suffix `.SchedulingObjective` and are stored in the `SchedulingObjective` folder.

Version

SchedulingObjective components are available in API version 55.0 and later.

Special Access Rules

This type is available only if Workforce Engagement is enabled in your org. To view, create, edit, and delete records, the user requires the Workforce Engagement Planner permission set.

Fields

Field Name	Description
`isProtected`	**Field Type** boolean

Field Name	Description
	Description Indicates whether the component is protected (`true`) or not (`false`). The default value is `false`.
`masterLabel`	**Field Type** string **Description** Required. The name of the objective.
`schedulingCategory`	**Field Type** SchedulingCategory (enumeration of type string) **Description** Required. What the scheduling logic applies the objective to. The valid values are: • `A`—Service Appointment • `B`—Shift
`schedulingObjectiveParameters`	**Field Type** `SchedulingObjectiveParameter[]` on page 1773 **Description** Parameters associated with a scheduling objective, such as the number of days before and after a shift that the logic considers when balancing assignments.
`schedulingObjectiveType`	**Field Type** SchedulingObjectiveType (enumeration of type string) **Description** Required. Specifies the type of objective. Possible values are: • `AgentPreference`—In the UI, this value appears as Maximized Preferences. • `BalanceNonStandardShifts` • `BalanceShifts`

SchedulingObjectiveParameter

Represents a parameter that's associated with a scheduling objective.

Field Name	Description
`parameterKey`	**Field Type** ObjectiveParameterKey (enumeration of type string) **Description** Required. The scheduling objective parameter key. Possible values are: • `DaysAhead`

Field Name	Description
	• DaysBack
value	**Field Type** string **Description** The scheduling objective parameter value.

Declarative Metadata Sample Definition

The following is an example of a SchedulingObjective component.

```
<?xml version="1.0" encoding="UTF-8"?>
<SchedulingObjective xmlns="http://soap.sforce.com/2006/04/metadata">
    <masterLabel>Balance Shifts</masterLabel>
    <schedulingCategory>B</schedulingCategory>
    <schedulingObjectiveType>BalanceShifts</schedulingObjectiveType>
    <schedulingObjectiveParameters>
        <parameterKey>DaysAhead</parameterKey>
        <value>30</value>
    </schedulingObjectiveParameters>
</SchedulingObjective>
```

The following is an example package.xml that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <name>SchedulingObjective</name>
 <members>Balance Shifts</members>
    </types>
    <version>55.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character * (asterisk) in the package.xml manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SchedulingRule

Represents a scheduling rule in Workforce Engagement Management. Scheduling rules determine when agents are assigned to shifts.

Parent Type

This type extends the Metadata metadata type and inherits its fullName field.

File Suffix and Directory Location

`SchedulingRule` components have the suffix `.schedulingRule` and are stored in the `SchedulingRules` folder.

Version

SchedulingRule components are available in API version 53.0 and later.

Special Access Rules

This type is available only if Workforce Engagement is enabled in your org. To view, create, edit, and delete records, the user requires the Workforce Engagement Planner permission set.

Fields

Field Name	Description
`isProtected`	**Field Type** boolean **Description** Indicates whether the component is protected (`true`) or not (`false`). The default value is `false`.
`masterLabel`	**Field Type** string **Description** Required. The name of the rule.
`schedulingCategory`	**Field Type** SchedulingCategory (enumeration of type string) **Description** Required. What the scheduling logic applies the rule to. The valid values are: • `A`—Service Appointment • `B`—Shift
`schedulingRuleParameters`	**Field Type** `SchedulingRuleParameter[]` on page 1776 **Description** Parameters associated with a scheduling rule, such as work limits.
`schedulingRuleType`	**Field Type** SchedulingRuleType (enumeration of type string) **Description** Required. Specifies the type of rule. The valid values are:

Field Name	Description
	• `A`—Active Resources • `B`—Match Skills • `C`—Availability • `M`—Match Territory • `Q`—Match Queue • `RestTimeMinutes`—Rest Time in Minutes. Available in API version 56.0 and later. • `W`—Work Limit • `LimitNonstandardShifts`—Specifies a rule type that limits how many non-standard shifts can be assigned to each agent. Available in API version 54.0 and later.

SchedulingRuleParameter

Represents a scheduling rule parameter, such as a work limit, that's associated with a scheduling rule.

Field Name	Description
`schedulingParameterKey`	**Field Type** SchedulingParameterKey (enumeration of type string) **Description** Required. The scheduling rule parameter key. • `C`—Constraint Field Name • `L`—Limit Type • `R`—Resolution • `T`—Time Resolution • `W`—Work Unit • `ConsiderAbsence`—Consider resource absences when evaluating availability. Available in API version 56.0 and later. • `ConsiderSTM`—Consider service territory membership, which defines working hours, when evaluating availability. Available in API version 56.0 and later.
`value`	**Field Type** string **Description** The scheduling rule parameter value.

Declarative Metadata Sample Definition

The following is an example of a `SchedulingRule` component.

```
<?xml version="1.0" encoding="UTF-8"?>
<SchedulingRule xmlns="http://soap.sforce.com/2006/04/metadata">
    <masterLabel>Max Shifts Per Week</masterLabel>
    <schedulingCategory>B</schedulingCategory>
    <schedulingRuleParameters>
        <schedulingParameterKey>C</schedulingParameterKey>
        <value>MaxShiftsPerWeek</value>
    </schedulingRuleParameters>
    <schedulingRuleParameters>
        <schedulingParameterKey>W</schedulingParameterKey>
        <value>Shifts</value>
    </schedulingRuleParameters>
    <schedulingRuleParameters>
        <schedulingParameterKey>R</schedulingParameterKey>
        <value>Week</value>
    </schedulingRuleParameters>
    <schedulingRuleParameters>
        <schedulingParameterKey>L</schedulingParameterKey>
        <value>Max</value>
    </schedulingRuleParameters>
    <schedulingRuleType>W</schedulingRuleType>
</SchedulingRule>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <name>SchedulingRule</name>
 <members>MaxShiftsPerWeek</members>
    </types>
    <version>53.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

Scontrol

Deprecated. Represents an Scontrol component, corresponding to an s-control in the Salesforce user interface.

Important: Visualforce pages supersede s-controls. Organizations that haven't previously used s-controls can't create them. Existing s-controls are unaffected and can still be edited.

This type extends the MetadataWithContent metadata type and inherits its `content` and `fullName` fields.

Declarative Metadata File Suffix and Directory Location

The file suffix is `.scf` for the s-control file. The accompanying metadata file is named *`ScontrolName`*`-meta.xml`.

Scontrol components are stored in the `scontrols` folder in the corresponding package directory.

Version

Scontrols are available in API version 10.0 and later.

Fields

This metadata type contains the following fields:

Field Name	Field Type	Description
`content`	base64Binary	Content of the s-control. Base 64-encoded binary data. Before making an API call, client applications must encode the binary attachment data as base64. Upon receiving a response, client applications must decode the base64 data to binary. This conversion is handled for you by a SOAP client. This field is inherited from the MetadataWithContent component.
`contentSource`	SControlContentSource (enumeration of type string)	Required. Determines how you plan to use the s-control: • `HTML`: Select this option if you want to enter the content for your s-control in `content`. • `URL`: Select this option if you want to enter the link or URL of an external website in `content`. • `Snippet`: Snippets are s-controls that are designed to be included in other s-controls. Select this option if you want to enter the content for your s-control snippet in `content`.
`description`	string	Optional text that describes the s-control. This only displays to users with View All Data permission (administrator).
`encodingKey`	Encoding (enumeration of type string)	Required. The default encoding setting is Unicode: `UTF-8`. Change it if you're passing information to a URL that requires data in a different format. This option is available when you select `URL` as the value for `contentSource`.
`fileContent`	base64	File contents displayed if you add this s-control to a custom link. The file can contain a Java applet, Active-X control, or any other type of content you want. This option only applies to s-controls with a value of `HTML` for `contentSource`.
`fileName`	string	The unique name for the s-control. This name can contain only underscores and alphanumeric characters, and must be unique in your org. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. This field can't be changed for components installed by a managed

Field Name	Field Type	Description
		package. It's only relevant if the `fileContent` field also has a value. This field is available in API version 14.0.
`fullName`	string	The s-control developer name used as a unique identifier for API access. The `fullName` can contain only underscores and alphanumeric characters. It must be unique, begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. If this field contained characters before version 14.0 that are no longer allowed, the characters were stripped out of this field, and the previous value of the field was saved in the name field. This field is inherited from the Metadata component.
`name`	string	Required. The unique name for the s-control. It must contain alphanumeric characters only and begin with a letter. For example `example_s_control`.
`supportsCaching`	boolean	Required. Indicates whether the s-control supports caching (`true`) or not (`false`). Caching optimizes the page so that it remembers which s-controls are on the page when it reloads. This option only applies to HTML s-controls.

Declarative Metadata Sample Definition

The following sample creates the `Myriad_Publishing.scf` s-control, which creates a link to the website specified in the s-control. The corresponding `Myriad_Publishing.scf-meta.xml` metadata file follows the s-control file.

`Myriad_Publishing.scf` file:

```
http://www.myriadpubs.com
```

`Myriad_Publishing.scf-meta.xml`:

```
<?xml version="1.0" encoding="UTF-8"?>
<Scontrol xmlns="http://soap.sforce.com/2006/04/metadata">
    <contentSource>URL</contentSource>
    <description>s-control to open Myriad Publishing website.</description>
    <encodingKey>UTF-8</encodingKey>
    <name>Myriad Publishing</name>
    <supportsCaching>true</supportsCaching>
</Scontrol>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SearchCustomization

Represents the configuration of search settings created in Search Manager. The configuration includes the search channel, searchable objects and fields, and rules to filter search results.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

SearchCustomization components have the suffix `.searchCustomization` and are stored in the `searchCustomizations` folder.

Version

SearchCustomization components are available in API version 61.0 and later.

Special Access Rules

Only users with the View Setup and Configuration permission can access this object, and only users with the Customize Application permission can edit it.

Fields

Field Name	Description
`channel`	**Field Type** string **Description** Required. The search channel that the configuration applies to.
`masterLabel`	**Field Type** string **Description** Required. The name of the configuration.
`objectOverride`	**Field Type** SearchCustomizationObjectOverride[]

Field Name	Description
	Description A list of object configurations.
`objectToAlwaysSearch`	**Field Type** string[] **Description** A list of the objects that are always searched for the user profile if the search channel is Einstein Global Search Bar.
`profile`	**Field Type** string **Description** Specifies user profile if the search channel is Einstein Global Search Bar.
`selectedObject`	**Field Type** string[] **Description** A list of the objects that are selected in the configuration if the search channel is LWR Experience Sites.
`selectedProfile`	**Field Type** string[] **Description** Specifies all user profiles that are associated with a Search configuration if the search channel is Einstein Global Search Bar. This field is available in API version 62.0 and later.

SearchCustomizationObjectOverride

Represents the configuration for a specific object.

Field Name	Description
`fieldOverride`	**Field Type** SearchCustomizationFieldOverride[] **Description** A list of field configurations.
`objectApiName`	**Field Type** string **Description** Required. The API name of the object that the configuration is applied to.

Field Name	Description
`rule`	**Field Type** SearchCustomizationRule[] **Description** A list of rules applied to filter search results.
`searchable`	**Field Type** boolean **Description** Indicates whether the object is searchable (`true`) or not (`false`).

SearchCustomizationFieldOverride

Represents the configuration for a specific field within an object.

Field Name	Description
`fieldApiName`	**Field Type** string **Description** Required. The API name of the field that the configuration is applied to.
`searchable`	**Field Type** boolean **Description** Required. Indicates whether the field is searchable (`true`) or not (`false`).

SearchCustomizationRule

Represents the rules defined in an object to filter search results.

Field Name	Description
`fieldApiName`	**Field Type** string **Description** Required. The field that the rule applies to.

Field Name	Description
operator	**Field Type** string **Description** Required. The operator for the rule.
ruleValue	**Field Type** SearchCustomizationRuleValue[] **Description** A list of rule values.

SearchCustomizationRuleValue

Represents the value of a rule used to filter search results.

Field Name	Description
targetObjectApiName	**Field Type** string **Description** The API name of the target object, in case the rule applies to a lookup field.
value	**Field Type** string **Description** Required. The value of the rule.

Declarative Metadata Sample Definition

The following is an example of a SearchCustomization component.

```
<?xml version="1.0" encoding="UTF-8"?>
<SearchCustomization xmlns="http://soap.sforce.com/2006/04/metadata">
    <channel>GlobalSearch</channel>
    <masterLabel>My_Standard_User_Configuration</masterLabel>
    <objectOverride>
        <fieldOverride>
            <fieldApiName>Description</fieldApiName>
            <searchable>false</searchable>
        </fieldOverride>
        <fieldOverride>
```

```
            <fieldApiName>Rating</fieldApiName>
            <searchable>true</searchable>
        </fieldOverride>
        <objectApiName>Account</objectApiName>
        <rule>
            <fieldApiName>My_Custom_Field__c</fieldApiName>
            <operator>ne</operator>
            <ruleValue>
                <value>Other</value>
            </ruleValue>
        </rule>
        <rule>
            <fieldApiName>Rating</fieldApiName>
            <operator>in</operator>
            <ruleValue>
                <value>Hot</value>
            </ruleValue>
            <ruleValue>
                <value>Warm</value>
            </ruleValue>
        </rule>
    </objectOverride>
    <objectOverride>
        <objectApiName>Asset</objectApiName>
        <searchable>false</searchable>
    </objectOverride>
    <objectOverride>
        <objectApiName>Contact</objectApiName>
        <rule>
            <fieldApiName>AccountId</fieldApiName>
            <operator>ne</operator>
            <ruleValue>
                <targetObjectApiName>Account</targetObjectApiName>
                <value>A Company</value>
            </ruleValue>
        </rule>
        <rule>
            <fieldApiName>DoNotCall</fieldApiName>
            <operator>eq</operator>
            <ruleValue>
                <value>false</value>
            </ruleValue>
        </rule>
    </objectOverride>
    <objectToAlwaysSearch>Account</objectToAlwaysSearch>
    <objectToAlwaysSearch>Contact</objectToAlwaysSearch>
    <objectToAlwaysSearch>My_Custom_Object__c</objectToAlwaysSearch>
    <objectToAlwaysSearch>Product2</objectToAlwaysSearch>
    <profile>standard</profile>
</SearchCustomization>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
```

```
    <types>
        <members>*</members>
        <name>SearchCustomization</name>
    </types>
    <version>61.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SearchOrgWideObjectConfig

Represents an object in the search index. The search index contains org-wide search settings created in Search Manager. Each object in the search index includes searchable fields and fields protected by field-level security in search.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

SearchOrgWideObjectConfig components have the suffix `.searchOrgWideObjectConfig` and are stored in the `searchOrgWideConfiguration` folder.

Version

SearchOrgWideObjectConfig components are available in API version 61.0 and later.

Special Access Rules

There are no additional access requirements that are specific to this type.

Fields

Field Name	Description
`masterLabel`	**Field Type** string **Description** Required.

Field Name	Description
	The name of the configuration.
`objectReference`	**Field Type** string **Description** Required. The API name of the object.
`searchOrgWideFieldConfig`	**Field Type** SearchOrgWideFieldConfig[] **Description** A list of field configurations.

SearchOrgWideFieldConfig

Represents the configuration in the search index for a field in an object.

Field Name	Description
`fieldReference`	**Field Type** string **Description** Required. The API name of the field.
`isSearchable`	**Field Type** boolean **Description** Indicates if the field is searchable (`true`) or not (`false`). If `true`, the field is shown in search results and used to match results.
`isSecure`	**Field Type** boolean **Description** Indicates if the field is protected by field-level security in search (`true`) or not (`false`). If `true`, the search engine uses this field to match results only for users with permissions. If `false`, the search engine uses this field to match results even if the user doesn't have permissions to view this field.

Declarative Metadata Sample Definition

The following is an example of a SearchOrgWideObjectConfig component.

```
<?xml version="1.0" encoding="UTF-8"?>
<SearchOrgWideObjectConfig xmlns="http://soap.sforce.com/2006/04/metadata">
    <masterLabel>CustomerLabel</masterLabel>
    <objectReference>Customer</objectReference>
    <searchOrgWideFieldConfig>
        <fieldReference>Custom_Field_1__c</fieldReference>
        <isSearchable>false</isSearchable>
        <isSecure>false</isSecure>
    </searchOrgWideFieldConfig>
    <searchOrgWideFieldConfig>
        <fieldReference>Custom_Field_2__c</fieldReference>
        <isSearchable>true</isSearchable>
        <isSecure>true</isSecure>
    </searchOrgWideFieldConfig>
</SearchOrgWideObjectConfig>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>SearchOrgWideObjectConfig</name>
    </types>
    <version>61.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ServiceAISetupDefinition

Represents settings for an Einstein for Service feature such as Einstein Article Recommendations. This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

ServiceAISetupDefinition components have the suffix `.serviceAISetupDescription` and are stored in the `serviceAISetupDescriptions` folder.

Version

ServiceAISetupDefinition components are available in API version 51.0 and later.

Special Access Rules

This type is available only when an org is configured to access the application in the `appSourceType` field. For example, if `appSourceType` is set to ARTICLE_RECOMMENDATION, this type is available only if Einstein Article Recommendations is enabled in the org and the Main Services Agreement has been accepted.

Fields

Field Name	Field Type	Description
`appSourceType`	ApplicationSourceType (enumeration of type string)	Required. The target application for the configuration. Valid values are: • `REPLY_RECOMMENDATION`—Einstein Reply Recommendations • `ARTICLE_RECOMMENDATION`—Einstein Article Recommendations • `UTTERANCE_RECOMMENDATION`—Einstein Bot utterances • `FAQ`—Einstein Bot frequently asked questions
`name`	string	Required. A reference to the configuration.
`setupStatus`	ServiceAISetupDefStatus (enumeration of type string)	Required. The status of the configuration. Valid values are: • `FIELDS_SELECTED` • `TRAINING` • `READY_TO_ACTIVATE` • `SERVING` • `RETIRED` • `ARCHIVED` • `READY_FOR_REVIEW`
`supportedLanguages`	string	Required when `appSourceType` is `ARTICLE_RECOMMENDATION`. Language codes for selected and supported languages.

Declarative Metadata Sample Definition

Here's an example of a ServiceAISetupDefinition component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceAISetupDefinition xmlns="http://soap.sforce.com/2006/04/metadata">
    <appSourceType>ARTICLE_RECOMMENDATION</appSourceType>
    <name>SA1601228426202</name>
    <setupStatus>ARCHIVED</setupStatus>
    <supportedLanguages>en,de,fr,it,es,pt,nl</supportedLanguages>
</ServiceAISetupDefinition>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>ServiceAISetupDefinition</name>
    </types>
</Package>
```

Usage

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ServiceAISetupField

Represents a field on cases or knowledge articles that Einstein uses to identify relevant articles in Einstein Article Recommendations. This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

ServiceAISetupField components have the suffix `.serviceAiSetupField` and are stored in the `serviceAiSetupFields` folder.

Version

ServiceAISetupField components are available in API version 51.0 and later.

Special Access Rules

This type is available only if Einstein Article Recommendations is enabled in your org and the Main Services Agreement has been accepted.

Fields

Field Name	Field Type	Description
`entity`	string	Required. The Case or KnowledgeArticle object for the field.
`field`	string	Required. The API name of the field.
`fieldMappingType`	ServiceAISetupFieldType (enumeration of type string)	Required. The field type. Valid values are: • `CASE_DESC` • `CASE_SUBJ` • `ARTICLE_TITLE` • `ARTICLE_CONTENT` • `ARTICLE_SUMMARY`

Field Name	Field Type	Description
`fieldPosition`	int	Required. A positive number used to rank the field's importance. The value 1 is most important; higher numbers indicate less important fields. Einstein considers fields in the order of importance.
`name`	string	Required. A reference to the field.
`setupDefinition`	string	Required. A reference to the parent ServiceAISetupDefinition.

Declarative Metadata Sample Definition

The following is an example of a ServiceAISetupField component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceAISetupField xmlns="http://soap.sforce.com/2006/04/metadata">
    <entity>Case</entity>
    <field>Subject</field>
    <fieldMappingType>CASE_SUBJ</fieldMappingType>
    <fieldPosition>1</fieldPosition>
    <name>SF16039900475920</name>
    <setupDefinition>4hQRM0000004CDK</setupDefinition>
</ServiceAISetupField>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>ServiceAISetupField</name>
    </types>
</Package>
```

Usage

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ServiceChannel

Represents a channel of work items that are received from your organization—for example, cases, chats, or leads.

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

ServiceChannel components have the suffix `.serviceChannel` and are stored in the `serviceChannels` folder.

Version

ServiceChannel components are available in API version 44.0 and later.

Special Access Rules

This type is available only if Omni-Channel is enabled in your org.

Fields

Field Name	Field Type	Description
`acwExtensionDuration`	int	The maximum length of time, measured in seconds, an agent can spend on After Conversation Work (ACW) each time they extend the timer. You must set this field if `hasAcwExtensionEnabled` is set to `true`. Specify a value from 10 through 3600. Available only for service channels of type Messaging or Voice.
`afterConvoMaxTime`	int	The maximum length of time, measured in seconds, an agent has to complete After Conversation Work (ACW). You must set this field if `hasAfterConvoWorkTimer` is set to `true`. Specify a value from 10 through 3600. Available only for service channels of type Messaging or Voice. For service channels of type Voice, this field is available in API version 52.0 and later. For service channels of type Messaging, this field is available in API version 56.0 and later.
`doesMinimizeWidgetOnAccept`	boolean	Automatically minimizes the Omni-Channel widget when an agent accepts work. This field is available in API version 48.0 and later.
`hasAcwExtensionEnabled`	Boolean	If set to `true`, agents can extend their After Conversation Work (ACW) time. Available only if `hasAfterConvoWorkTimer` is set to `true`. If set to `true`, you must also set the `acwExtensionDuration` and `maxExtensions` fields. The default value is `false`. Available only for service channels of type Messaging or Voice. This field is available in API version 56.0 and later.
`hasAfterConvoWorkTimer`	Boolean	If set to `true`, After Conversation Work (ACW) time can be configured for the channel. If set to `true`, you must also set the `afterConvoWorkMaxTime` field. The default value is `false`. Available only for service channels of type Messaging or Voice. For service channels of type Voice, this field is available in API version 52.0 and later. For service channels of type messaging, this field is available in API version 56.0 and later.
`hasAutoAcceptEnabled`	Boolean	Work items in a service channel open automatically in the agent's workspace so that the agent doesn't have to manually accept them.

Field Name	Field Type	Description
interactionComponent	string	The custom console component to open in the footer when an agent accepts a work item from this service channel.
isInterruptible	boolean	Indicates whether a work item consumes interruptible or primary capacity. The default value is false. Available in API version 57.0 and later when the Interruptible Capacity feature is enabled.
label	string	Required. The label of the service channel.
maxExtensions	picklist	The maximum number of times an agent can extend their After Work Conversation (ACW) time. Specify a value from 1 through 10. You must set this field if `hasAcwExtensionEnabled` is set to `true`. Available only for service channels of type Messaging or Voice. This field is available in API version 56.0 and later.
relatedEntityType	string	Required. The type of object that's associated with this service channel.
secondaryRoutingPriorityField	string	The name of the standard field or the ID of the custom field that is used for secondary routing priority. This field is available in API version 47.0 and later.
serviceChannelFieldPriorities	ServiceChannelFieldPriority[]	Required. A set of mappings between secondary routing priority field values and priorities. This field is available in API version 47.0 and later.

ServiceChannelFieldPriority

Represents a secondary routing priority field value mapping. Available in API version 47.0 and later.

Field Name	Field Type	Description
priority	int	Required. The priority number assigned to the mapped field value.
value	string	Required. The value of the secondaryRoutingPriorityField field defined in the parent ServiceChannel.

Declarative Metadata Sample Definition

The following is an example of a ServiceChannel component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceChannel xmlns="http://soap.sforce.com/2006/04/metadata">
    <doesMinimizeWidgetOnAccept>true</doesMinimizeWidgetOnAccept>
    <interactionComponent>ConsoleComponent</interactionComponent>
    <label>Case</label>
    <relatedEntityType>Case</relatedEntityType>
    <secondaryRoutingPriorityField>Status</secondaryRoutingPriorityField>
    <serviceChannelFieldPriorities>
        <priority>1</priority>
        <value>Escalated</value>
    </serviceChannelFieldPriorities>
```

```
    <serviceChannelFieldPriorities>
        <priority>2</priority>
        <value>On Hold</value>
    </serviceChannelFieldPriorities>
</ServiceChannel>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>ServiceChannel</name>
    </types>
    <version>44.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ServicePresenceStatus

Represents a presence status that can be assigned to a service channel. This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

ServicePresenceStatus components have the suffix `.servicePresenceStatus` and are stored in the `servicePresenceStatuses` folder.

Version

ServicePresenceStatus components are available in API version 44.0 and later.

Special Access Rules

This type is available only if Omni-Channel is enabled in your org.

Fields

Field Name	Field Type	Description
`channels`	ServiceChannelStatus	Represents the status that's associated with a specific service channel.
`label`	string	The label of the presence status.

ServiceChannelStatus

Represents the status that's associated with a specific service channel.

Field Name	Field Type	Description
`channel`	string	Represents the channels assigned to the presence status.

Declarative Metadata Sample Definition

The following is an example of a ServicePresenceStatus component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ServicePresenceStatus xmlns="http://soap.sforce.com/2006/04/metadata">
    <channels>
        <channel>Case</channel>
    </channels>
    <label>Available for Cases</label>
</ServicePresenceStatus>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>ServicePresenceStatus</name>
    </types>
    <version>44.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ServiceProcess

Represents a process created in Service Process Studio and its associated attributes.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

ServiceProcess components have the suffix `.serviceprocess` and are stored in the `.serviceprocess` folder.

Version

ServiceProcess components are available in API version 57.0 and later.

Special Access Rules

Access to the ServiceProcess type requires the AccessToServiceProcess permission.

Fields

Field Name	Description
`description`	**Field Type** string **Description** A meaningful explanation of the service process.
`processLabel`	**Field Type** string **Description** Required. A meaningful name for the service process.
`serviceProcessAttributes`	**Field Type** ServiceProcessAttribute[] **Description** Custom attributes that store the data associated with the service process.
`serviceProcessDependencies`	**Field Type** ServiceProcessDependency[] **Description** Dependent components of the service process, such as OmniScripts or flows.
`serviceProcessItemGroups`	**Field Type** ServiceProcessItemGroup[] **Description** Groups of related ServiceProcessAttribute records.
`shortDescription`	**Field Type** string **Description** A brief meaningful explanation of the service process.

Field Name	Description
`usageType`	**Field Type** SvcCatalogItemUsageType (enumeration of type string) **Description** Required. The Cloud that uses this service process. Values are: • `CustomerService` • `Employee` • `FinancialServices` • `Industry (available in version 58.0 and later)`

ServiceProcessAttribute

A custom attribute that stores data associated with a service process. For example, a service process that reverses a fee can have a Fee Type attribute.

Field Name	Description
`attributeType`	**Field Type** SvcCtlgItemAttrAttributeType (enumeration of type string) **Description** A `Base` attribute corresponds to a SvcCatalogRequest field, which is subject to field-level security. An `Extended` attribute is only a ServiceProcessAttribute object record, which isn't subject to field-level security. Values are: • `Base` • `Extended` The default is `Extended`.
`dataType`	**Field Type** SvcCatalogItemAttrDataType (enumeration of type string) **Description** The data type of the attribute. Values are: • `Checkbox` • `Currency` • `Date` • `Datetime` • `Integer`

Field Name	Description
	• `ListOfBoolean` • `ListOfDouble` • `ListOfInteger` • `ListOfMaps` • `ListOfString` • `Map` • `Number` • `Percentage` • `Text` The default is `Text`. Note: Selecting `Currency` doesn't cause an error, but currency conversions aren't supported.
`description`	**Field Type** string **Description** A meaningful explanation of the attribute.
`developerName`	**Field Type** string **Description** Required. A system name for the attribute.
`fieldIdentifier`	**Field Type** string **Description** For a `Base` attribute, the Developer Name of the SvcCatalogRequest field. This field can be standard or custom.
`groupApiName`	**Field Type** string **Description** The `apiName` of the ServiceProcessItemGroup to which this attribute belongs.
`inputVariableValue`	**Field Type** string **Description** The default value of the attribute.

Field Name	Description
`isRequired`	**Field Type** boolean **Description** Specifies whether the attribute is required. The default is `false`.
`label`	**Field Type** string **Description** Required. A meaningful name for the attribute.
`parentAttribute`	**Field Type** string **Description** The parent attribute of this attribute. For example, a Latitude attribute can have GeoLocation as a parent.
`sortOrder`	**Field Type** int **Description** The position of the attribute in the payload relative to other attributes having no parent or the same parent.

ServiceProcessDependency

A dependent component of the service process, which can be a flow, an OmniScript, an Integration Definition, or a preprocessor Apex class.

Field Name	Description
`dependencyReference`	**Field Type** string **Description** Required. The Developer Name of the flow, OmniScript, Integration Definition, or preprocessor Apex class.
`processStepName`	**Field Type** SvcCtlgItemDpndProcType (enumeration of type string) **Description** Name of the step in a service process.

Field Name	Description
	Values are: • `FulfillmentFlow` • `IntegrationDefinition` • `Preprocessor` • `RequestForm`
`type`	**Field Type** SvcCatalogItemDependencyType (enumeration of type string) **Description** Required. The type of dependent component. Values are: • `FlowDefinition` • `IntegrationProviderDef` • `OmniScriptConfig` • `PreprocessorApexClass`

ServiceProcessItemGroup

A group of related ServiceProcessAttribute records.

Field Name	Description
`apiName`	**Field Type** string **Description** Required. The API Name of the group.
`groupName`	**Field Type** string **Description** Required. The name of the group.
`sortOrder`	**Field Type** int **Description** Required.

Field Name	Description
	The group display order.

Declarative Metadata Sample Definition

The following is an example of a ServiceProcess component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceProcess xmlns="http://soap.sforce.com/2006/04/metadata">
   <processLabel>EmailUpdate</processLabel>
   <usageType>FinancialServices</usageType>
   <serviceProcessAttributes>
      <label>EmailAddress</label>
      <developerName>EmailAddress</developerName>
      <dataType>Text</dataType>
      <groupApiName>Info</groupApiName>
   </serviceProcessAttributes>
   <serviceProcessDependencies>
      <dependencyReference>EmailPreprocessor</dependencyReference>
      <type>PreprocessorApexClass</type>
   </serviceProcessDependencies>
   <serviceProcessItemGroups>
        <apiName>Info</apiName>
        <groupName>Info</groupName>
        <sortOrder>1</sortOrder>
    </serviceProcessItemGroups>
</ServiceProcess>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>ServiceProcess</name>
    </types>
    <version>57.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

Usage Type

We recommend that you review these considerations before you retrieve or deploy service process metadata.

- If you deploy metadata with the same name as the definition when your service process definition is active, you get an error message. Deactivate the service process definition and try again.
- When your service process definition is inactive, consider these guidelines.

- If a service process definition contains service catalog requests and service catalog request extended attribute values and you deploy metadata with the same name as the definition, you get an error message. You can't delete or change a service process that has service catalog requests with attribute values in it. Make sure that all records are deleted in service catalog requests and service catalog request extended attribute values before you deploy the metadata.
- If a service process definition contains service catalog requests but doesn't contain service catalog request extended attribute values and you deploy the metadata with the same name, the deployment works as expected.
- If a service process definition doesn't contain service catalog requests and you deploy the metadata with the same name, the deployment works as expected.

Settings

Represents the organization settings related to a feature. For example, your password policies, session settings and network access controls are all available in the SecuritySettings component type.

Not all feature settings are available in the Metadata API. See Unsupported Metadata Types on page 160 for information on which feature settings are not available.

Settings can be accessed using the specific component member or via wildcard. For example, in the package manifest file you would use the following section to access SecuritySettings:

```
    <types>
        <members>Security</members>
        <name>Settings</name>
    </types>
```

The member format when used in the package manifest is the component metadata type name without the "Settings" suffix, so in the preceding example "Security" is used instead of "SecuritySettings".

File Suffix and Directory Location

Each settings component gets stored in a single file in the `settings` directory of the corresponding package directory. The filename uses the format *`Setting feature`*`.settings`. For example, the SecuritySettings file would be `Security.settings`. See "File Suffix and Directory Location" information for the individual settings components to determine the exact filename.

Version

Settings is available in API version 27.0 and later. See the version information for the individual setting component to determine which API version the settings component became available.

Declarative Metadata Sample Definition

The following is an example package manifest used to deploy or retrieve only the MobileSettings for an organization:

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>Mobile</members>
        <name>Settings</name>
    </types>
```

```
    <version>27.0</version>
</Package>
```

The following is an example package manifest used to deploy or retrieve all the available settings metadata for an organization, using a wildcard:

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>Settings</name>
    </types>
    <version>27.0</version>
</Package>
```

AccountSettings

Represents an org's account settings for account teams, account owner report, and the **View Hierarchy** link.

AccountInsightsSettings

Represents an org's Einstein Account Insights settings. This setting controls features that help your reps maintain their relationships with their customers.

AccountIntelligenceSettings

Represents an org's Account Intelligence settings. These settings control features that make it easy for sales reps to create accounts, see relevant news articles, and add logos to account records. This type extends the Metadata metadata type and inherits its `fullName` field.

AccountingSettings

Represents the settings for the Accounting Subledger feature.

ActionsSettings

Represents an org's actions settings for default quick actions, multi-dimensional publisher, and third-party actions. This type extends the Metadata metadata type and inherits its `fullName` field.

ActivitiesSettings

Represents an org's activity settings, and its user interface settings for the calendar. This type extends the Metadata metadata type and inherits its `fullName` field.

AddressSettings

Represents the configuration of country/territory and state picklists. Use the AddressSettings component type to configure state and country/territory data in your organization so that you can convert text-based values into standard picklist values. To convert your state and country/territory values, from Setup, enter *`State and Country/Territory Picklists`* in the Quick Find box, then select **State and Country/Territory Picklists**.

AIReplyRecommendationsSettings

Represents the metadata used to manage settings for Einstein Reply Recommendations. This type extends the Metadata metadata type and inherits its `fullName` field.

AgentforceForDevelopersSettings

Represents Agentforce for Developers settings.

AnalyticsSettings

Represents Analytics settings in Salesforce. CRM Analytics lets you explore all your data quickly and easily by providing AI-powered advanced Analytics right inside Salesforce. Manage your datasets, query data with Salesforce Analytics Query Language (SAQL), and customize dashboards. You can use these settings to configure which Analytics features are available to users in your organization.

ApexSettings

Represents Apex-related org settings. This type extends the Metadata metadata type and inherits its `fullName` field.

AppAnalyticsSettings

Represents settings to retrieve AppExchange App Analytics usage data.

AppExperienceSettings

Represents settings for the app experience.This type extends the Metadata metadata type and inherits its `fullName` field.

AssociationEngineSettings

Represents the record association builder settings for an org. This type extends the Metadata metadata type and inherits its `fullName` field.

AutomatedContactsSettings

Represents an org's Einstein Automated Contacts settings. These settings let you find new contacts and opportunity contact roles. This type extends the Metadata metadata type and inherits its `fullName` field.

BotSettings

Represents an organization's Einstein Bot settings, such as whether or not Einstein Bots is enabled. This type extends the Metadata metadata type and inherits its `fullName` field.

BranchManagementSettings

Represents the branch management settings for an org. This type extends the Metadata metadata type and inherits its `fullName` field.

BusinessHoursSettings

Represents the metadata used to manage settings for business hours and holidays in entitlements, entitlement templates, campaigns, and cases. This type extends the Metadata metadata type and inherits its `fullName` field.

CampaignSettings

Represents an org's Campaign Influence, Einstein Attribution, Einstein Key Accounts, and campaign member settings. These features help you understand how your campaigns and accounts are affecting your opportunity pipeline.

CaseSettings

Represents an organization's case settings, such as the default case owner, which case-related features are enabled, and which Classic email templates are used for various case activities. This type extends the Metadata metadata type and inherits its `fullName` field.

ChatterAnswersSettings

Represents the metadata used to manage settings for Chatter Answers.

ChatterEmailsMDSettings

Represents an org's settings for Chatter email when Chatter is enabled. This type extends the Metadata metadata type and inherits its `fullName` field.

ChatterSettings

Represents an org's settings for their Chatter instance when Chatter is enabled for the org. This type extends the Metadata metadata type and inherits its `fullName` field.

CodeBuilderSettings

Represents Code Builder settings. This type extends the Metadata metadata type and inherits its `fullName` field.

CollectionsDashboardSettings

Represents an org's settings to add the Collections Dashboard application to an org.

CommunitiesSettings

Represents community settings for an org. Enable digital experiences and workspaces. Manage moderation, guest user and partner settings, and more. This type extends the Metadata metadata type and inherits its `fullName` field.

CompanySettings

Represents global settings that affect multiple features in your organization. This type extends the Metadata metadata type and inherits its `fullName` field.

ConnectedAppSettings

Represents settings for connected apps. This type extends the Metadata metadata type and inherits its `fullName` field.

ContentSettings

Represents content settings for an org. This type extends the Metadata metadata type and inherits its `fullName` field.

ContractSettings

Represents contract settings.

ConversationalIntelligenceSettings

Represents the org's Einstein Conversation Insights settings, such as whether Einstein Conversation Insights is enabled. Einstein Conversation Insights lets you analyze your rep's call recordings, and gives you the insights you need to optimize every call.

ConversationChannelDefinition

Represents the conversation channel definition that's implemented for Interaction Service for Bring Your Own Channel for Messaging and Bring Your Own Channel for CCaaS messaging channels. This object is available in API version 60.0 and later.

CurrencySettings

Represents an organization's currency settings, including supporting multiple currencies and currency effective dates. This type extends the Metadata metadata type and inherits its `fullName` field.

CustomAddressFieldSettings

Represents the settings for custom address fields.

DataDotComSettings

Represents the org's Data.com settings. This type extends the Metadata metadata type and inherits its `fullName` field.

DataImportManagementSettings

Represents an org's contact and leads import settings.

DeploymentSettings

Represents the settings affecting how deployments behave in the org. This type extends the Metadata metadata type and inherits its `fullName` field.

DevHubSettings

Represents Dev Hub settings.

DocumentGenerationSetting

Represents an org's settings for automatic document generation from templates. This type extends the Metadata metadata type and inherits its `fullName` field.

DynamicFormsSettings

Represents the settings related to Dynamic Forms.

EACSettings

Represents the Einstein Activity Capture metadata type. Use Einstein Activity Capture to add emails and events from your Microsoft or Google account to the activity timeline of related Salesforce records. Automatically sync contact and event data between your Microsoft or Google account and Salesforce. This type extends the Metadata metadata type and inherits its `fullName` field.

EinsteinAgentSettings

Represents settings for Einstein classification apps, Einstein Case Classification and Einstein Case Wrap-Up, in an org. This type extends the Metadata metadata type and inherits its `fullName` field.

EinsteinGptSettings

Represents settings for Einstein Generative AI features in an org. This type extends the Metadata metadata type and inherits its `fullName` field

EmailAdministrationSettings

Represents an organization's email administration settings, including email deliverability, security compliance, relay configurations, and system notifications. This type extends the Metadata metadata type and inherits its `fullName` field.

EmailIntegrationSettings

Represents an org's settings for the Outlook integration, Gmail integration, and Salesforce Inbox. This type extends the Metadata metadata type and inherits its `fullName` field.

EmailTemplateSettings

Represents an org's email template settings. This type extends the Metadata metadata type and inherits its `fullName` field.

EmployeeUserSettings

Represents the employee-user settings used for automatically creating or syncing employee and user data in work.com orgs. This type extends the Metadata metadata type and inherits its `fullName` field.

EnhancedNotesSettings

Represents an org's enhanced note settings, such as enabling enhanced notes and enabling tasks in enhanced notes.This type extends the Metadata metadata type and inherits its `fullName` field.

EncryptionKeySettings

Represents an org's encryption key settings, such as customer-supplied keys options and key derivation settings. This type extends the Metadata metadata type and inherits its `fullName` field.

EntitlementSettings

Represents an organization's entitlement settings.

EventSettings

Represents an org's platform event settings for Event Monitoring.

ExperienceBundleSettings

Represents the org setting that enables the ExperienceBundle metadata type for Aura sites in Experience Cloud. The setting doesn't affect LWR sites, which use ExperienceBundle by default. This type extends the Metadata metadata type and inherits its `fullName` field.

ExternalClientAppSettings

Represents settings to enable the External Client App feature and provide access to the OAuth consumer secret.

ExternalServicesSettings

Represents settings for an External Services registration.

FieldServiceSettings

Represents an organization's Field Service settings.

FilesConnectSettings

Represents the settings that modify the Files Connect feature.This type extends the Metadata metadata type and inherits its `fullName` field.

FileUploadAndDownloadSecuritySettings

Represents the security settings for uploading and downloading files. This type extends the Metadata metadata type and inherits its `fullName` field.

FlowSettings

Represents the Salesforce settings for processes and flows, such as whether Lightning runtime for flows is enabled.

ForecastingObjectListSettings

Represents an org's forecasting object list settings. Use these settings to control which object types and field types appear in the list of object details on the forecasts page. For example, pipeline forecasts use the Opportunity object, and the object list settings specify which fields from that object are available in the opportunity list section of the forecasts page. This type extends the Metadata metadata type and inherits its `fullName` field.

ForecastingSettings

Represents the Forecasts settings options. This type extends the Metadata metadata type and inherits its `fullName` field.

HighVelocitySalesSettings

Represents an org's Sales Engagement settings. With Sales Engagement, you can make your inside sales team as effective as possible.

IdeasSettings

Represents the metadata used to manage settings for Ideas.

IdentityProviderSettings

Represents the settings used to enable or disable Salesforce as a SAML identity provider for single sign-on (SSO).

IframeWhiteListUrlSettings

Represents settings related to the list of trusted external domains that you allow to frame your Visualforce pages or surveys. This type extends the Metadata metadata type and inherits its `fullName` field.

IncidentMgmtSettings

Represents settings for Customer Service Incident Management and Broadcast Communications.

IndustriesEinsteinFeatureSettings

Represents the settings for enabling the Industries Einstein feature.

IndustriesLoyaltySettings

Represents the settings to enable capabilities of Loyalty Management.

IndustriesSettings

Represents settings for industries verticals such as Financial Services Cloud, Consumer Goods Cloud, Public Sector Solutions, Education Cloud, Salesforce Scheduler, Life Sciences Cloud, and Health Cloud.

InterestTaggingSettings

Represents settings for Interest Tags, which your users can add to client records to capture client needs, interests, and prospecting opportunities.

InventorySettings

Represents options for the Salesforce Omnichannel Inventory product.This type extends the Metadata metadata type and inherits its `fullName` field.

InvLatePymntRiskCalcSettings

Represents the org's settings to identify the level of risks associated with payment of invoices.

InvocableActionSettings

Represents the org's invocable action settings, such as whether partial save is allowed.This type extends the Metadata metadata type and inherits its `fullName` field.

KnowledgeSettings

Represents the metadata used to manage settings for Salesforce Knowledge.

LanguageSettings

Represents an organization's language settings. Language settings control end-user language selection, locale formats, and translation options. This type extends the Metadata metadata type and inherits its `fullName` field.

LeadConfigSettings

Represents configuration settings for Leads that control how they are converted and displayed, and what actions are available. This type extends the Metadata metadata type and inherits its `fullName` field.

LeadConvertSettings

Represents an organization's custom field mappings for lead conversion. Custom fields can be mapped from Leads to Accounts, Contacts, and Opportunities. Options for creating opportunities during lead conversion can also be specified. This type extends the Metadata metadata type and inherits its `fullName` field.

LiveAgentSettings

Represents an organization's Chat settings, such as whether Chat is enabled. This type extends the Metadata metadata type and inherits its `fullName` field.

LightningExperienceSettings

Represents the settings that modify an org's Lightning Experience configuration. This type extends the Metadata metadata type and inherits its `fullName` field.

LiveMessageSettings

Represents an org's LiveMessage settings.

MacroSettings

Represents an organization's Macro settings, such as whether or not folders is enabled. This type extends the Metadata metadata type and inherits its `fullName` field.

MailMergeSettings

Represents the settings for Extended Mail Merge functionality.

MapAndLocationSettings

Represents an org's map and location settings.

MeetingsSettings

Represents the settings to enable Salesforce Meetings and the integration with Zoom video conferencing.

MobileSettings

Represents an organization's mobile settings. This type extends the Metadata metadata type and inherits its `fullName` field.

MyDomainSettings

Represents your org's My Domain settings. With My Domain, you can include your company name in your URLs, for example, `https://yourcompanyname.my.salesforce.com`. This type extends the Metadata metadata type and inherits its `fullName` field.

MfgServiceConsoleSettings

Represents the settings to access the Service Console for Manufacturing.This type extends the Metadata metadata type and inherits its `fullName` field.

NameSettings

Enables or disables the formal name, middle name, and suffix attributes for these person objects: Contact, Lead, Person Account, and User. This type extends the Metadata metadata type and inherits its `fullName` field.

NotificationsSettings

Represents an organization's mobile settings.

OauthOidcSettings

Represents org settings for disabling OAuth OpenID Connect authorization flows.

ObjectHierarchyRelationship

Represents an organization's custom field mappings for sales agreement conversion. Fields can be mapped from Opportunity and Quotes to SalesAgreement and SalesAgreementProduct.

ObjectLinkingSettings (Beta)

Represents the channel-object linking settings for an org. This type extends the Metadata metadata type and inherits its `fullName` field.

OmniChannelSettings

Represents the Omni-Channel settings for an org.

OpportunityInsightsSettings

Represents an org's Einstein Opportunity Insights settings. This setting controls features that give you relevant updates about your opportunities.

OpportunitySettings

Represents org preferences for features such as automatic opportunity updates and similar-opportunity filters.

OpportunityScoreSettings

Represents an org's Einstein Opportunity Scoring settings, such as whether or not Einstein Opportunity Scoring is enabled. Einstein Opportunity Scoring helps determine the likelihood of an opportunity being won. This type extends the Metadata metadata type and inherits its `fullName` field.

OrderManagementSettings

Represents options for the Salesforce Order Management product. This type extends the Metadata metadata type and inherits its `fullName` field.

OrderSettings

Represents order settings.

OrgPreferenceSettings

Removed in API version 48.0. Represents the unique org preference settings in a Salesforce org.

OrgSettings

Represents the settings for org-wide functionality that isn't associated with any specific feature.This type extends the Metadata metadata type and inherits its `fullName` field.

PartyDataModelSettings

Represents an organization's party data model settings, including options around the Individual object and consent enablement. This type extends the Metadata metadata type and inherits its `fullName` field.

PardotSettings

Represents Marketing Cloud Account Engagement settings in your Salesforce org. Account Engagement, formerly known as Pardot, is a B2B marketing automation solution that helps you create meaningful connections, generate more pipeline, and close more deals. Use these settings to configure how Account Engagement collects and displays data.

PardotEinsteinSettings

Represents PardotEinsteinSettings. Use these settings to learn what factors drive your campaign performance, and get the best possible engagement score for your prospects. This type extends the Metadata metadata type and inherits its `fullName` field.

PathAssistantSettings

Represents the Path preference setting. This type extends the Metadata metadata type and inherits its `fullName` field.

PaymentsSettings

Represents the Salesforce Payments settings when this feature is enabled for the org.

PicklistSettings

Represents an org's picklist settings. These settings control the behavior of a picklist. This type extends the Metadata metadata type and inherits its `fullName` field.

PlatformEncryptionSettings

Represents an org's Platform Encryption settings, such as settings for available encryption schemes, permissions, encryption policy access, and which fields can be encrypted. This type extends the Metadata metadata type and inherits its `fullName` field.

PlatformEventSettings

Represents settings for platform events and change data capture events.

PredictionBuilderSettings

Represents the settings that determine how a user can interact with Einstein Prediction Builder. This type extends the Metadata metadata type and inherits its `fullName` field.

PrivacySettings

Represents an organization's settings for data privacy and consent management. This type extends the Metadata metadata type and inherits its `fullName` field.

ProcessFlowMigration

Represents a process's migrated criteria and the resulting migrated flow.

ProductSettings

Represents organization preferences for quantity schedules, revenue schedules, and active flag interaction with prices. This type extends the Metadata metadata type and inherits its `fullName` field.

QuoteSettings

Represents an org's quotes settings, such as enabling quotes or creating quotes without an associated opportunity. This type extends the Metadata metadata type and inherits its `fullName` field.

RealTimeEventSettings

Represents the list of Real-Time Event entities that you want to enable or disable. This type extends the Metadata metadata type and inherits its `fullName` field.

RecordPageSettings

Represents an org's record page settings. This type extends the Metadata metadata type and inherits its `fullName` field.

RetailExecutionSettings

Represents settings to manage your inventory, promotions, planograms, and in-store activities.

SalesAgreementSettings

Represents settings that control the display of agreement terms metrics in sales agreements and the calculation of the actual quantity of products in sales agreements. These settings also control the approval of sales agreements.

SandboxSettings

Represents Sandbox settings. This type extends the Metadata metadata type and inherits its `fullName` field.

SchemaSettings

Represents an org's schema settings, which manage the availability of custom settings and custom metadata type values. This type extends the Metadata metadata type and inherits its `fullName` field.

SearchSettings

Represents an org's search settings.

SecuritySettings

Represents an org's security settings. For example, settings define trusted IP ranges for network access, password and login requirements, session expiration, and single sign-on settings.

ServiceCloudVoiceSettings

Represents an organization's Service Cloud Voice settings. This type extends the Metadata metadata type and inherits its `fullName` field.

ServiceSetupAssistantSettings

Represents an organization's Service Setup Assistant settings. The Service Setup Assistant can be used to set up a basic service console app.

SharingSettings

Represents an organization's sharing, visibility, and data access settings. This type extends the Metadata metadata type and inherits its `fullName` field.

SiteSettings

Represents the settings for Experience Cloud sites and for Salesforce Sites.

SocialCustomerServiceSettings

Represents Social Customer Service settings such as how to format inbound content from social posts to cases. This type extends the Metadata metadata type and inherits its `fullName` field.

SocialProfileSettings

Represents org preferences for social media features such as enabling Twitter and Facebook.Represents org preferences for social media features such as enabling Twitter and Facebook. This type extends the Metadata metadata type and inherits the fullName field.

SourceTrackingSettings (Beta)

Represents settings for source tracking, so that changes you make in your Developer and Developer Pro sandboxes or local workspace can be tracked. This type extends the Metadata metadata type and inherits its `fullName` field.

SubscriptionManagementSettings

Represents the settings used to manage recurring subscriptions.

SurveySettings

Represents an org's survey settings. Use the SurveySettings component to enable Salesforce Surveys, enable Customer Lifecycle Maps, and choose whether the owner of a survey can manage the responses.

Territory2Settings

Represents an org's Territory2 settings. Use Territory2 settings to set the access level that Sales Territories users have to records associated with sales territories, and to enable features. The standard record access settings apply to accounts and opportunities. With *`Private`* default internal access for contacts or cases, you can also set access for those records.This type extends the Metadata metadata type and inherits its `fullName` field.

TrailheadSettings

Represents an org's integration with Trailhead for Learning Paths or Enablement programs, including access to enablement sites (formerly myTrailhead).

TrialOrgSettings

Represents the settings in a trial user's org. This type extends the Metadata metadata type and inherits its `fullName` field.

UserEngagementSettings

Represents the metadata associated with various feature settings around Lightning Experience transition and adoption, user engagement and adoption assistance, and adoption apps.

UserInterfaceSettings

Represents the settings that modify the behavior of the org's user interface.

UserManagementSettings

Represents a selection of user management options that appear on the User Management Settings Setup page. This type extends the Metadata metadata type and inherits its `fullName` field.

VoiceSettings

Represents an org's Sales Dialer settings, such as call recording, conferencing, and voicemail.

WarrantyLifeCycleMgmtSettings

Represents settings that control the Warranty Administration for your org.

WorkDotComSettings

Represents WorkDotCom settings. This type extends the Metadata metadata type and inherits its `fullName` field.

WorkforceEngagementSettings

Represents settings for Workforce Engagement Management.

AccountSettings

Represents an org's account settings for account teams, account owner report, and the **View Hierarchy** link.

This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

AccountSettings values are stored in the `Account.settings` file in the `settings` folder. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

AccountSettings is available in API versions 29.0 and later.

Fields

Field Name	Field Type	Description
`enableAccountDiscovery`	boolean	When `true`, sets up Einstein Account Management dashboards and installs the related CRM Analytics and Customer Insights apps. The dashboards give users access to account health analytics including metrics on open pipeline, risk, and engagement scores. Einstein Account Management is part of Revenue Intelligence, which is available for an additional cost. Available in API version 57.0 and later.

Field Name	Field Type	Description
enableAccountHistoryTracking	boolean	Indicates whether history tracking is enabled for accounts (true) or not (false). The default value is false. If history tracking is disabled, the History related list is removed from account page layouts. However, history data is still available for reporting up to the date and time when tracking was disabled. Available in API version 47.0 and later.
enableAccountInsightsInMobile	boolean	Deprecated in API version 59.0 and later because the feature is no longer available. Indicates whether users can see Einstein Account Insights on their mobile device (true) or not (false). Insights appear in the Einstein Insights component, which is on account records and the Home page. To use this feature, users must have the Einstein Account Insights permission. Available in API version 47.0 to 58.0.
enableAccountOwnerReport	boolean	Indicates whether the Account Owner Report can (true) or can't (false) be run by all users.
enableAccountTeams	boolean	Indicates whether account teams are enabled (true) or not (false). The Metadata API can't be used to disable account teams.
enableContactHistoryTracking	boolean	Indicates whether history tracking is enabled for contacts (true) or not (false). Available in API version 46.0 and later.
enableRelateContactToMultipleAccounts	boolean	Indicates whether users can relate a contact to multiple accounts (true) or only one account (false). The default value is false. If this feature (Contacts to Multiple Accounts) is disabled, secondary contact–account relationships created while the feature was enabled are deleted. Available in API version 47.0 and later. Avoid using the Metadata API to enable this feature. Use the Account Settings page in Setup to enable Contacts to Multiple Accounts.
enableReportsToOnPersonAccount	boolean	Indicates whether the **Reports To** field on Person Account, which corresponds to the PersonReportsToId field on the Account object, is enabled. The field allows users to associate person accounts and contacts with other person accounts or contacts that they report to. Available in API version 62.0 and later.
showViewHierarchyLink	boolean	Indicates whether the default **View Hierarchy** link on all business account detail pages is visible (true) or hidden (false).

Declarative Metadata Sample Definition

The following is an example of the Account.settings file:

```
<?xml version="1.0" encoding="UTF-8"?>
<AccountSettings xmlns="http://soap.sforce.com/2006/04/metadata">
 <enableAccountDiscovery>true</enableAccountDiscovery>
 <enableAccountHistoryTracking>true</enableAccountHistoryTracking>
 <enableAccountInsightsInMobile>false</enableAccountInsightsInMobile>
 <enableAccountOwnerReport>true</enableAccountOwnerReport>
 <enableAccountTeams>true</enableAccountTeams>
 <enableContactHistoryTracking>true</enableContactHistoryTracking>
 <enableRelateContactToMultipleAccounts>true</enableRelateContactToMultipleAccounts>
 <enableReportsToOnPersonAccount>true</enableReportsToOnPersonAccount>
 <showViewHierarchyLink>true</showViewHierarchyLink>
</AccountSettings>
```

Example Package Manifest

The following is an example package manifest used to deploy or retrieve the Account settings metadata:

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
 <types>
  <members>Account</members>
  <name>Settings</name>
 </types>
 <version>29.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

AccountInsightsSettings

Represents an org's Einstein Account Insights settings. This setting controls features that help your reps maintain their relationships with their customers.

Note: This metadata type has been deprecated as of API version 59.0.

This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

AccountInsightsSettings values are stored in the `AccountInsights.settings` file in the `settings` folder. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

AccountInsightsSettings is available in API versions 48.0 to 58.0.

Fields

Field Name	Field Type	Description
`enableAccountInsights`	boolean	Indicates whether Einstein Account Insights is enabled (`true`) or not (`false`). The default value is `false`.

Declarative Metadata Sample Definition

The following is an example of the AccountInsights.settings file:

```
<?xml version="1.0" encoding="UTF-8"?>
<AccountInsightsSettings xmlns="http://soap.sforce.com/2006/04/metadata">
 <enableAccountInsights>true</enableAccountInsights>
</AccountInsightsSettings>
```

Example Package Manifest

The following is an example package manifest used to deploy or retrieve the AccountInsights settings metadata:

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
 <types>
  <members>AccountInsights</members>
  <name>Settings</name>
 </types>
 <version>29.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

AccountIntelligenceSettings

Represents an org's Account Intelligence settings. These settings control features that make it easy for sales reps to create accounts, see relevant news articles, and add logos to account records. This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

AccountIntelligenceSettings values are stored in the `AccountIntelligence.settings` file in the `settings` folder. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

AccountIntelligenceSettings is available in API versions 48.0 and later.

Fields

Field Name	Field Type	Description
`enableAccountLogos`	boolean	Indicates whether your sales reps can see available company logos (`true`) or not (`false`). The logos are for US-based companies only. The default value is `false`. `enableAutomatedAccountFields` must be `true` to use this setting.
`enableAutomatedAccountFields`	boolean	Indicates whether Automated Account Fields is enabled (`true`) or not (`false`). The default value is `false`.
`enableNewsStories`	boolean	Indicates whether News is enabled (`true`) or not (`false`). The default value is `false`. `enableAutomatedAccountFields` must be `true` to use this setting.

Declarative Metadata Sample Definition

The following is an example of the AccountIntelligence.settings file:

```
<?xml version="1.0" encoding="UTF-8"?>
<AccountIntelligenceSettings xmlns="http://soap.sforce.com/2006/04/metadata">
 <enableAccountLogos>true</enableAccountLogos>
 <enableAutomatedAccountFields>true</enableAutomatedAccountFields>
 <enableNewsStories>true</enableNewsStories>
</AccountIntelligenceSettings>
```

Example Package Manifest

The following is an example package manifest used to deploy or retrieve the AccountIntelligence settings metadata:

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
 <types>
  <members>AccountIntelligence</members>
  <name>Settings</name>
 </types>
```

```
 <version>29.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

AccountingSettings

Represents the settings for the Accounting Subledger feature.

Parent Type and Manifest Access

This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all the settings metadata types for the org are accessed using the "Settings" name. See Settings for more details.

File Suffix and Directory Location

`AccountingSettings` values are stored in the `AccountingSettings.settings` file in the `settings` folder. The `.settings` files are different from other named components, because there is only one settings file for each settings component.

Version

AccountingSettings components are available in API version 57.0 and later.

Fields

Field Name	Description
`enableAccountingSubledger`	**Field Type** boolean **Description** Indicates whether Transaction Journal creation is enabled for the organization (`true`) or not (`false`).
`enableFinancePeriod`	**Field Type** boolean **Description** Reserved for internal use.
`enablePaymentMethodAdjust`	**Field Type** boolean

Field Name	Description
	Description Indicates whether changes to the Payment Method generate adjustments on Transaction Journal records (`true`) or not (`false`).
`enableScheduledJob`	**Field Type** boolean **Description** Reserved for internal use.
`enableSkipReversalLogicEnabled`	**Field Type** boolean **Description** Indicates whether Skip Reversal Logic is enabled (`true`) or not (`false`).

Declarative Metadata Sample Definition

The following is an example of an AccountingSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<AccountingSettings
 xmlns="http://soap.sforce.com/2006/04/metadata">
 <enableAccountingSubledger>true</enableAccountingSubledger>
 <enablePaymentMethodAdjust>true</enablePaymentMethodAdjust>
</AccountingSettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package
 xmlns="http://soap.sforce.com/2006/04/metadata">
 <types>
  <members>Accounting</members>
  <name>Settings</name>
 </types>
 <version>57.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ActionsSettings

Represents an org's actions settings for default quick actions, multi-dimensional publisher, and third-party actions. This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

ActionsSettings values are stored in the `Actions.settings` file in the `settings` folder. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

Components are available in API version 47.0 and later.

Fields

Field Name	Field Type	Description
`enableDefaultQuickActionsOn`	boolean	Indicates whether default quick actions are created in the org (`true`, the default setting) or not (`false`).
`enableMdpEnabled`	boolean	Indicates whether multi-dimensional publisher is enabled (`true`, the default setting) or not (`false`).
`enableThirdPartyActions`	boolean	Indicates whether third-party actions are displayed in the multi-dimensional publisher (`true`) or not (`false`, the default setting).
`enableOfflineWebLinks`	boolean	Indicates whether a button or link is available offline (`true`), or if it's only available online (`false`, the default setting).

Declarative Metadata Sample Definition

The following is an example of an ActionsSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ActionsSettings xmlns="http://soap.sforce.com/2006/04/metadata">
 <DefaultQuickActionsOn>true</DefaultQuickActionsOn>
 <MdpEnabled>true</MdpEnabled>
 <ThirdPartyActions>true</ThirdPartyActions>
</ActionsSettings>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ActivitiesSettings

Represents an org's activity settings, and its user interface settings for the calendar. This type extends the Metadata metadata type and inherits its `fullName` field.

Use the ActivitiesSettings component type to control the following activity settings:

- Configure group and recurring tasks, recurring and multiday events, and email tracking
- Relate multiple contacts to tasks and events (shared activities)
- Display custom logos in meeting requests

Also use the ActivitiesSettings component type to control user interface settings for the calendar, including hover links and drag-and-drop editing.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

ActivitiesSettings values are stored in the `Activities.settings` file in the `settings` directory. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

ActivitiesSettings is available in API versions 28.0 and later.

Fields

Settings for all types listed below are controlled on the Activity Settings page or the User Interface settings page as noted.

Field Name	Field Type	Description
allowUsersToRelateMultipleContactsToTasksAndEvents	boolean	This field indicates whether Shared Activities is enabled. When the value is true, allows users to relate multiple contacts to a task or event. **Important:** Beginning with API v36.0, this field is read-only in all versions of the API. You can't change the value of this field. Even though this field was updateable before Spring '16, changing this field's value wasn't supported and could have resulted in an incorrect integration. If you have code in older API versions that changes the value of this field, ensure that you update that code to prevent any errors.
`autoRelateEventAttendees`	boolean	When users add attendees to events, events are automatically related to up to 50 contacts or one lead. An attendee is matched by their email address to a contact or lead. Admins control this field on the Activity Settings page. Available in API version 42.0 and later.
`enableActivityReminders`	boolean	Enables popup activity reminders for an organization. Admins control this field on the Activity Settings page.

Field Name	Field Type	Description
`enableCalendarHomeLWC`	boolean	Enables Lightning Web Components for Calendar. Increases the default item limit in Calendar Home and applies styling enhancements to improve readability. Admins control this field on the Activity Settings page.
`enableClickCreateEvents`	boolean	Lets users create events in day and weekly calendar views by double-clicking a specific time slot and entering the details of the event in an overlay. Hovering over an event displays an overlay where users can view the event details or delete the event without leaving the page. Admins use a mini page layout to configure the fields shown in the overlays. Doesn't support recurring events or multi-person events. Admins control this field on the User Interface settings page.
`enableDragAndDropScheduling`	boolean	Lets users create events associated with records by dragging a record from a list view onto a calendar view and entering the details of the event in an overlay. Hovering over an event displays an overlay where users can view the event details or delete the event without leaving the page. Admins use a mini page layout to configure the fields shown in the overlays. Admins control this field on the User Interface settings page.
`enableEmailTracking`	boolean	Enables tracking of outbound HTML emails if an organization uses HTML email templates. Admins control this field on the Activity Settings page.
`enableFlowTaskNotifsViaApex`	boolean	If Apex invokes Process Builder to create a task, determines whether an email is sent (`true`) or not (`false`).
`enableGroupTasks`	boolean	Lets users assign independent copies of a new task to multiple users. Admins control this field on the Activity Settings page.
`enableHideChildEventsPreference`	boolean	Enables hiding child events from the calendar or activity views. This setting is useful if you have complex event hierarchies and want to simplify the views by hiding less relevant details. This field is available in API version 50.0 and later. Admins control this field on the Activity Settings page.
`enableListViewScheduling`	boolean	Extends the functionality of `enableDragAndDropScheduling` and `enableClickCreateEvents` to list view calendars. Admins control this field on the User Interface settings page.
`enableLogNote`	boolean	Enables the option to create and associate a note on an existing record.
`enableMLSingleClientProfile`	boolean	Enable creating a client profile using machine learning. When this setting is enabled, Salesforce uses machine learning algorithms to analyze and consolidate client data, providing a more comprehensive client profile.

Field Name	Field Type	Description
		This can help sales and service teams to better understand their clients and provide more personalized interactions. This field is available in API version 50.0 and later. Admins control this field on the Activity Settings page.
`enableMultidayEvents`	boolean	Enables creation of events that end more than 24 hours after they start. Admins control this field on the Activity Settings page.
`enableRecurringEvents`	boolean	Enables creation of events that repeat at specified intervals. Admins control this field on the Activity Settings page.
`enableRecurringTasks`	boolean	Enables creation of tasks that repeat at specified intervals. Admins control this field on the Activity Settings page.
`enableRollUpActivToContactsAcct`	boolean	Enables a contact's activities to be rolled up and displayed on the contact's primary account. Default value is `true`. Available in API versions 47.0 and later.
`enableSidebarCalendarShortcut`	boolean	In the sidebar, displays a shortcut link to a user's last-used calendar view. Admins control this field on the Activity Settings page.
`enableSimpleTaskCreateUI`	boolean	Allows admins to specify whether tapping New Task in Salesforce opens a regular task record edit page or a page that displays key task fields first. Admins control this field on the Activity Settings page.
`enableTimelineCompDateSort`	boolean	Allows admins to sort past activities by completed date (`true`). If `false`, activities are sorted by due date. Admins control this field on the Activity Settings page.
`enableUNSTaskDelegatedToNotifications`	boolean	On the Activity Settings page, exposes a setting for Admins to hide or show a user setting that lets individual users enable or disable email notifications when tasks are assigned to them.
`enableUserListViewCalendars`	boolean	Allows users to create and view user list view calendars in Lightning Experience. Available in API versions 47.0 and later
`meetingRequestsLogo`	string	Available when `showCustomLogoMeetingRequests` is enabled. Uploads a custom logo. An administrator can select only a logo that has been uploaded to certain folders in the Documents tab. Admins control this field on the Activity Settings page.
`showCustomLogoMeetingRequests`	boolean	Displays a custom logo in meeting request emails and on a meeting's Web page. Invitees see the logo when a user either invites them to an event or requests a meeting. Admins control this field on the Activity Settings page.

Field Name	Field Type	Description
showEventDetailsMultiUserCalendar	boolean	Displays event details on-screen rather than in hover text. Admins control this field on the Activity Settings page.
showHomePageHoverLinksForEvents	boolean	In the calendar section of the Home tab: • When a user hovers over the subject of an event, a hover link displays an overlay with selected event details. (Hover links are always available in other calendar views.) • When a user clicks the subject of an event, displays the event detail page. Admins use a mini page layout to configure the fields shown in the overlay. Admins control this field on the User Interface settings page.
showMyTasksHoverLinks	boolean	In the My Tasks section of the Home tab and on the calendar day view: • When a user hovers over the subject of a task, a hover link displays an overlay with selected task details. • When a user clicks the subject of a task, displays the task detail page. Admins use a mini page layout to configure the fields shown in the overlay. Admins control this field on the User Interface settings page.

Example Package Manifest

The following is an example package manifest used to deploy or retrieve the activity settings metadata for an organization:

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>Activities</members>
        <name>Settings</name>
    </types>
    <version>28.0</version>
</Package>
```

Declarative Metadata Sample Definition

The following is an example of an activity settings file:

```
<?xml version="1.0" encoding="UTF-8"?>
<ActivitiesSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <enableActivityReminders>true</enableActivityReminders>
    <autoRelateEventAttendees>true</autoRelateEventAttendees>
    <enableClickCreateEvents>true</enableClickCreateEvents>
    <enableDragAndDropScheduling>true</enableDragAndDropScheduling>
    <enableEmailTracking>true</enableEmailTracking>
```

```
    <enableGroupTasks>true</enableGroupTasks>
    <enableListViewScheduling>true</enableListViewScheduling>
    <enableMultidayEvents>true</enableMultidayEvents>
    <enableRecurringEvents>true</enableRecurringEvents>
    <enableRollUpActivToContactsAcct>true</enableRollUpActivToContactsAcct>
    <enableRecurringTasks>true</enableRecurringTasks>
    <enableTimelineCompDateSort>true</enableTimelineCompDateSort>
    <enableUserListViewCalendars>true</enableUserListViewCalendars>
    <enableSidebarCalendarShortcut>true</enableSidebarCalendarShortcut>
    <meetingRequestsLogo>Folder02/logo03.png</meetingRequestsLogo>
    <showCustomLogoMeetingRequests>true</showCustomLogoMeetingRequests>
    <showEventDetailsMultiUserCalendar>true</showEventDetailsMultiUserCalendar>
    <showHomePageHoverLinksForEvents>true</showHomePageHoverLinksForEvents>
    <showMyTasksHoverLinks>true</showMyTasksHoverLinks>
</ActivitiesSettings>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SEE ALSO:

Document

AddressSettings

Represents the configuration of country/territory and state picklists. Use the AddressSettings component type to configure state and country/territory data in your organization so that you can convert text-based values into standard picklist values. To convert your state and country/territory values, from Setup, enter *`State and Country/Territory Picklists`* in the Quick Find box, then select **State and Country/Territory Picklists**.

This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

Declarative Metadata File Suffix and Directory Location

AddressSettings values are stored in a single file named `Address.settings` in the `settings` directory. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

AddressSettings is available in API versions 27.0 and later.

CountriesAndStates

This complex metadata type represents valid definitions of states and countries/territories in picklists.

Note: You can use the Metadata API to edit existing states, countries, and territories in state and country/territory picklists. You can't use the Metadata API to create or delete new states, countries, or territories.

Field	Field Type	Description
`countries`	Country[]	The countries and territories available in picklists.

Country

This metadata type provides the definition for a country/territory in a picklist.

Field	Field Type	Description
`active`	boolean	Determines whether the value is available in the API. **Important:** After you enable state and country/territory picklists in your Salesforce organization, you can't set the `active` status to *`false`*.
`integrationValue`	string	A customizable text value that is linked to a state or country/territory code. Integration values for standard states, countries, and territories default to the full ISO-standard state, country, and territory names. Integration values function similarly to the API names of custom fields and objects. Configuring integration values allows integrations that you set up before enabling state and country/territory picklists to continue to work. **Important:** If you don't specify integration values before enabling state and country/territory picklists in your organization, records use the default value provided by Salesforce. If you change integration values later, records created or updated from that point on use your edited values.
`isoCode`	string	The ISO-standard code populates this field when you issue a `retrieve()` call. This field is read only in the API but you can edit the label in Setup. You can't edit the `isoCode` of `standard` states, countries, and territories.
`label`	string	The label is what users see in picklists in Salesforce. This field is read only in the API but you can edit the label in Setup.
`orgDefault`	boolean	Sets a country or territory as the default value for new records in the Salesforce organization.
`standard`	boolean	Standard states and countries are states and countries that are included with Salesforce. You can't edit the `standard` attribute.
`states`	State[]	The states or provinces that are part of the country or territory.
`visible`	boolean	Makes the state, country, or territory available to users in Salesforce. States, countries, or territories that are `visible` must also be `active`.

State

This metadata type provides the definition for a state in a picklist.

Field	Field Type	Description
`active`	boolean	Determines whether the value is available in the API. **Important:** After you enable state and country/territory picklists in your Salesforce organization, you can't set the `active` status to *`false`*.
`integrationValue`	string	A customizable text value that is linked to a state or country/territory code. Integration values for standard states, countries, and territories default to the full ISO-standard state, country, and territory names. Integration values function similarly to the API names of custom fields and objects. Configuring integration values allows integrations that you set up before enabling state and country/territory picklists to continue to work. **Important:** If you don't specify integration values before enabling state and country/territory picklists in your organization, records use the default value provided by Salesforce. If you change integration values later, records created or updated from that point on use your edited values.
`isoCode`	string	The ISO-standard code populates this field when you issue a `retrieve()` call. This field is read only in the API but you can edit the label in Setup.
`label`	string	The label is what users see in picklists in Salesforce. This field is read only in the API but you can edit the label in Setup.
`standard`	boolean	Standard states and countries are states and countries that are included with Salesforce. You can't edit the `standard` attribute.
`visible`	boolean	Makes the state, country, or territory available to users in Salesforce. States, countries, or territories that are `visible` must also be `active`.

Declarative Metadata Sample Definition

The following is sample XML that configures state and country picklists for the United States and Canada for use in an organization. It also makes the country of Greenland available only in the API. This example is supported in API version 64.0.

```
<?xml version="1.0" encoding="UTF-8"?>
<AddressSettings xmlns="http://soap.sforce.com/2006/04/metadata">
  <countriesAndStates>
    <countries>
      <country>
```

```
    <active>true</active>
    <integrationValue>United States</integrationValue>
    <isoCode>US</isoCode>
    <label>United States</label>
    <orgDefault>true</orgDefault>
    <standard>true</standard>
    <states>
      <state>
        <active>true</active>
        <integrationValue>Alabama</integrationValue>
        <isoCode>AL</isoCode>
        <label>Alabama</label>
        <standard>true</standard>
        <visible>true</visible>
      </state>
      <state>
        <active>true</active>
        <integrationValue>Alaska</integrationValue>
        <isoCode>AK</isoCode>
        <label>Alaska</label>
        <standard>true</standard>
        <visible>true</visible>
      </state>
    </states>
    <visible>true</visible>
  </country>
  <country>
    <active>true</active>
    <integrationValue>Canada</integrationValue>
    <isoCode>CA</isoCode>
    <label>Canada</label>
    <orgDefault>false</orgDefault>
    <states>
      <state>
        <active>true</active>
        <integrationValue>Alberta</integrationValue>
        <isoCode>AB</isoCode>
        <label>Alberta</label>
         <standard>true</standard>
         <visible>true</visible>
      </state>
      <state>
        <active>true</active>
        <integrationValue>British Columbia</integrationValue>
        <isoCode>BC</isoCode>
        <label>British Columbia</label>
        <standard>true</standard>
        <visible>true</visible>
      </state>
    </states>
    <visible>true</visible>
  </country>
  <country>
    <active>true</active>
```

```
        <integrationValue>Greenland</integrationValue>
        <isoCode>GL</isoCode>
        <label>Greenland</label>
        <standard>true</standard>
        <visible>false</visible>
      </country>
    </countries>
  </countriesAndStates>
</AddressSettings>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

AIReplyRecommendationsSettings

Represents the metadata used to manage settings for Einstein Reply Recommendations. This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

Einstein Reply Recommendations settings are stored in a single file named `aireplyrecommendations.settings` in the `settings` folder. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

AIReplyRecommendationsSettings is available in API version 49.0 and later.

Fields

Field Name	Field Type	Description
`enableAIReplyRecommendations`	boolean	If `true` (default), Einstein Reply Recommendations is enabled. If `false`, it is disabled.
`enableGenReplyRecommendations`	boolean	If `true` (default), Einstein Service Replies is enabled. If `false`, it is disabled. Available in API version 58.0 or later.
`enableServiceEinsteinGPTGrounding`	boolean	If `true` (default), Service AI Grounding is enabled. If `false`, it is disabled. Available in API version 58.0 or later.

Declarative Metadata Sample Definition

The following is an example aireplyrecommendations.settings metadata file:

```
<?xml version="1.0" encoding="UTF-8"?>
<AIReplyRecommendationsSettings xmlns="http://soap.sforce.com/2006/04/metadata">
 <enableAIReplyRecommendations>true</enableAIReplyRecommendations>
</AIReplyRecommendationsSettings>
```

Example Package Manifest

The following is an example package.xml manifest that references the AIReplyRecommendationsSettings definitions:

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
 <types>
  <members>AIReplyRecommendations</members>
  <name>Settings</name>
 </types>
 <version>49.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

Copyright

Rights of ALBERT EINSTEIN are used with permission of The Hebrew University of Jerusalem. Represented exclusively by Greenlight.

AgentforceForDevelopersSettings

Represents Agentforce for Developers settings.

Parent Type and Manifest Access

This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all org settings metadata types are accessed using the Settings name. See Settings for more details.

File Suffix and Directory Location

AgentforceForDevelopersSettings values are stored in the `AgentforceForDevelopers.settings` file in the `settings` folder. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

`AgentforceForDevelopersSettings` are available in API versions 62.0 and later.

Fields

Field Name	Field Type	Description
agentforceForDevelopersOptOut	boolean	Indicates whether Agentforce for Developers is enabled: `true` or `false`. If `true`, Agentforce for Developers isn't enabled in your org, which means that you've opted out of using it. If `false`, Agentforce for Developers is enabled. The default value is `false`.

Declarative Metadata Sample Definition

The following is an example of the `AgentforceForDevelopers.settings` file:

```
<?xml version="1.0" encoding="UTF-8"?>
        <AgentforceForDevelopersSettings xmlns="http://soap.sforce.com/2006/04/metadata">

        <agentforceForDevelopersOptOut>false</agentforceForDevelopersOptOut>
        </AgentforceForDevelopersSettings>
```

The following is an example `package.xml` manifest that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
     <types>
         <members>AgentforceForDevelopersSettings</members>
         <name>Settings</name>
     </types>
     <version>62.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

AnalyticsSettings

Represents Analytics settings in Salesforce. CRM Analytics lets you explore all your data quickly and easily by providing AI-powered advanced Analytics right inside Salesforce. Manage your datasets, query data with Salesforce Analytics Query Language (SAQL), and customize dashboards. You can use these settings to configure which Analytics features are available to users in your organization.

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

AnalyticsSettings values are stored in the `Analytics.settings` file in the `settings` folder. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

AnalyticsSettings components are available in API version 46.0 and later.

Special Access Rules

The AnalyticsSettings metadata type is accessible in all organizations. The fields that pertain to Reports and Dashboards are available in all organizations, but fields that pertain to CRM Analytics are only available in organizations with CRM Analytics enabled.

Fields

Field Name	Field Type	Description
`alwaysGenPreviews`	boolean	Indicates whether Analytics asset previews are generated (`true`) or not (`false`). Available in API version 47.0 and later.
`analyticsAdoptionMetadata`	boolean	Indicates whether Adoption Analytics metadata collection can be installed via a dataflow in Salesforce (`true`) or not (`false`). Available in API version 47.0 and later.
`analyticsCalendarApp`	boolean	Indicates whether the Analytics Calendar app for Industry templates can be installed in Salesforce (`true`) or not (`false`). Available in API version 49.0. Removed in API version 50.0.
`autoInstallApps`	boolean	Indicates whether CRM Analytics apps can be auto-installed in Salesforce (`true`) or not (`false`). Available in API version 54.0 and later.
`bundleCachingOptOut`	boolean	Indicates whether the default CRM Analytics dashboard bundle caching behavior is disabled (`true`) or enabled (`false`). Available in API version 58.0 and later.
`canAccessAnalyticsViaAPI`	boolean	Indicates whether Analytics assets can be accessed via the Analytics REST API in Salesforce (`true`) or not (`false`). Available in API version 47.0 and later.
`canAnnotateDashboards`	boolean	Indicates whether the Analytics dashboards Chatter annotation feature is enabled in Salesforce (`true`) or not (`false`). Available in API version 47.0 and later.
`canEnableBYOMZeroDay Scoring`	boolean	Indicates whether zero day scoring on user uploaded Einstein Discover model is enabled in Salesforce (`true`) or not (`false`). Available in API version 54.0 to 56.0. Removed in API version 57.0.
`canEnableLiveMetrics`	boolean	Indicates whether the Data Discovery live model metrics calculation feature is enabled in Salesforce (`true`) or not (`false`). Available in API version 48.0 and 49.0. Removed in API version 50.0.

Field Name	Field Type	Description
canEnableSavedView	boolean	Indicates whether the saved view feature for Analytics dashboards is enabled in Salesforce (true) or not (false). Available in API version 47.0 and later.
canExploreData Conversationally	boolean	Indicates whether Analytics data can be explored via NLQ (true) rather than using strict SAQL statements (false). For example, "Show me all accounts that are closed won". Available in API version 47.0 and later.
canShareAppsWith Communities	boolean	Indicates whether Analytics apps can be shared with Experience Builder sites and their users, outside of the standard Analytics Studio experience (true) or not (false). Available in API version 47.0 and later.
canSubscribeDashboard Widgets	boolean	Indicates whether a user can subscribe to Analytics dashboard widgets in Salesforce (true) or not (false). Available in API version 47.0 to 50.0. Removed in API version 51.0.
canViewThumbnailAssets	boolean	Indicates whether the thumbnail representations of Analytics lenses and dashboards are viewable (true) or not (false). Available in API version 47.0 and later.
concurrencyLimitSharing	boolean	Indicates whether the concurrency limits of Data Prep dataflows and recipes can be shared (true) or not (false). Available in API version 60.0 and later.
enableAmazonRedshift OutputConnector	boolean	Indicates whether the Amazon Redshift Output connector is enabled in Salesforce (true) or not (false). Available in API version 58.0 and later.
enableAnalyticsEncryption	boolean	Indicates whether encryption is enabled for Analytics in Salesforce (true) or not (false). Available in API version 48.0 and later.
enableAnalyticsSharing Enable	boolean	Indicates whether the Analytics sharing is enabled in Salesforce (true) or not (false). Available in API version 48.0 and later.
enableAutoCompleteCombo	boolean	Indicates whether using auto-complete when choosing reports and dashboards is enabled in Salesforce (true) or not (false). Available in API version 47.0 and later.
enableAutonomous Experience	boolean	Indicates whether Ask Salesforce for Data is enabled in Salesforce (true) or not (false). Available as Beta in API version 54.0 and later.
enableAzureDLGen2Output Connector	boolean	Indicates whether the Azure DL Gen2 output connector is enabled in Salesforce (true) or not (false). Available in API version 54.0 and later.

Field Name	Field Type	Description
`enableC360GlobalProfileData`	boolean	Indicates whether the Customer 360 data validation dashboard connector is enabled in Salesforce (`true`) or not (`false`). Available in API version 48.0 or later.
`enableCreateLegacyDataflows`	boolean	Indicates whether access to creating dataflows is available in Salesforce (`true`) or not (`false`). Available in API version 55.0 and later.
`enableCrmaDataCloudIntegration`	boolean	Indicates whether platform integration between CRM Analytics and Data Cloud is enabled (`true`) or not (`false`). Available in API version 58.0 and later.
`enableCrtSetupLightningUiPref`	boolean	Indicates whether the enhanced custom report type setup pages is enabled in Salesforce (`true`) or not (`false`). Available in API version 64.0 and later.
`enableDashboardChangeOwnerPref`	boolean	Indicates whether changing ownership of Lightning Experience dashboards from one owner to another owner is enabled in Salesforce (`true`) or not (`false`). Available in API version 59.0. Removed in API version 60.0.
`enableDashboardCmpRefreshPref`	boolean	Indicates whether Lightning Experience dashboard component refresh is enabled in Salesforce (`true`) or not (`false`). Available in API version 64.0 and later.
`enableDashboardComponentSnapshot`	boolean	Indicates whether posting dashboard component snapshots to feeds that are visible to all users is enabled in Salesforce (`true`) or not (`false`). Available in API version 47.0 and later.
`enableDashboardFlexiTable`	boolean	Indicates whether access is enabled to flexible dashboard tables for all users in Salesforce (`true`) or not (`false`). Available in API version 47.0 and later.
`enableDashboardToPDFEnable`	boolean	Indicates whether a dashboard can be exported to a PDF in Salesforce (`true`) or not (`false`). Available in API version 48.0 and later.
`enableDashboardSubOrgEmailPref`	boolean	Indicates whether the unique org email for Lightning Experience dashboard subscriptions is enabled (`true`) or not (`false`). Available in API version 64.0 and later.
`enableDataCloudReportingPref`	boolean	Indicates whether Data Cloud reporting with the analytics query engine is enabled (`true`) or not (`false`). Available in API version 59.0 and later.

Field Name	Field Type	Description
enableDataBlending	boolean	Indicates whether the Analytics Explorer data blending feature is available in Salesforce (true) or not (false). Available in API version 48.0 and 49.0. Removed in API version 50.0.
enableEmailReportsToPortalUsers	boolean	Indicates whether this org allows Classic reports and dashboards to be sent to Portal Users (true) or not (false). Available in API version 47.0 and later.
enableFirebirdEditor	boolean	Indicates whether the Firebird editor is available in Salesforce (true) or not (false). Available in API version 48.0 and later.
enableFloatingReportHeaders	boolean	Indicates whether report results display floating headers when scrolling (true) or not (false). Available in API version 47.0 and later.
enableIncludeDisclaimerMessage	boolean	Indicates whether confidental data disclaimers are included with analytics assets (true) or not (false). Available in API version 64.0 and later.
enableInsights	boolean	Indicates whether CRM Analytics is enabled in Salesforce (true) or not (false).
enableInsightsHCMode	boolean	Indicates whether CRM Analytics for Public Cloud is enabled in Salesforce (true) or not (false). Available in API version 58.0 and later.
enableLightningReportBuilder	boolean	Indicates whether the Lightning Report Builder feature can be enabled or disabled on the Setup page in Salesforce (true) or not (false). Available in API version 47.0 and later.
enableLotusNotesImages	boolean	Indicates whether the use of Lotus Notes-friendly images in dashboards and report emails is available in Salesforce (true) or not (false). Available in API version 47.0 and later.
enableLwcInDashboards	boolean	Indicates whether Lightning Web Components are enabled for use in CRM Analytics Dashboards (true) or not (false). Available as Beta in API version 53.0. Removed for GA in API version 54.0.
enableMassEnableReportBuilder	boolean	Indicates whether the Report Builder is available in Salesforce (true), overriding profile level settings, or not (false). Available in API version 47.0 and later.
enableNewChartsEngine	boolean	Indicates whether the New Charts Engine for reports and dashboards is available in Salesforce (true) or not (false). Available in API version 47.0 and later.

Field Name	Field Type	Description
`enableNullDimension`	boolean	Indicates whether null values are supported as a grouping key value in a SAQL query in Salesforce (`true`) or not (`false`). Available in API version 48.0 and later.
`enableOrgCanSeeLive Previews`	boolean	Indicates whether admins can enable live previews of data in Salesforce (`true`) or not (`false`). Available in API version 57.0 and later.
`enableOrgCanViewTableau`	boolean	Indicates whether admins can enable Tableau dashboards in Salesforce (`true`) or not (`false`). Available in API version 55.0 and later.
`enableOrgCanViewThumbnail ForOA`	boolean	Indicates whether admins can enable thumbnails for Lightning Experience reports and dashboards in Salesforce (`true`) or not (`false`). Available in API version 57.0 and later.
`enableOrgHaMobileOffline Enabled`	boolean	Indicates whether admins can enable mobile offline access in Salesforce (`true`) or not (`false`). Available in API version 51.0 and later.
`enableOrgHasWatchlist Enabled`	boolean	Indicates whether admins can turn on watchlists for assets in Salesforce (`true`) or not (`false`). Available in API version 50.0 and later.
`enablePowerInsights`	boolean	Indicates whether admins can turn on Power Insights for this org (`true`) or not (`false`). Removed in API version 51.0.
`enableQueryLiveConnectors`	boolean	Indicates whether querying live connectors is available in Salesforce (`true`) or not (`false`). Available in API version 48.0 and later.
`enableRecommendedReport TypePref`	boolean	Indicates whether recommended report types for Lightning Reports are available in Salesforce (`true`) or not (`false`). Available in API version 54.0 to 56.0. Removed in API version 57.0.
`enableRemoveFooterForRep Display`	boolean	Indicates whether the default disclaimer for the report run page and printable view page is removed (`true`) or not (`false`) in Salesforce. Available in API version 47.0 and later.
`enableRemoveFooterFromRep Exp`	boolean	Indicates whether the default footer from the exported (csv/excel) report is removed (`true`) or not (`false`) in Salesforce. Available in API version 47.0 and later.
`enableReportCdnPref`	boolean	Indicates whether the content delivery network (CDN) feature in Lightning Experience Reports is available in Salesforce (`true`) or not (`false`). Available in API version 64.0 and later.

Field Name	Field Type	Description
`enableReportFieldToFieldPref`	boolean	Indicates whether the field-to-field filters feature in Lightning Experience Reports is available in Salesforce (`true`) or not (`false`). Available in API version 47.0. Removed in API version 48.0.
`enableReportCrtAutoAddPref`	boolean	Indicates whether the feature to automatically add new fields to relevant custom Lightning Experience report types when they're created is available in Salesforce (`true`) or not (`false`). Available in API version 50.0 and 51.0. Removed in API version 52.0.
`enableReportHideXlsExportPref`	boolean	Indicates whether the XLS export feature for Lightning Experience Reports is visible in Salesforce (`true`) or not (`false`). Available in API version 51.0 and later. Note: To manage this setting, users must have the Export Reports user permission.
`enableReportInlineEditPref`	boolean	Indicates whether the inline editing feature for Lightning Experience Reports is available in Salesforce (`true`) or not (`false`). Available in API version 53.0 and later.
`enableReportNotificationsEnable`	boolean	Indicates whether the notification feature for Lightning Experience Reports is available in Salesforce (`true`) or not (`false`). Available in API version 48.0 and later.
`enableReportSubOrgEmailPref`	boolean	Indicates whether the unique org email for Lightning Experience Report subscriptions is enabled (`true`) or not (`false`). Available in API version 61.0 and later.
`enableReportUniqueRowCountPref`	boolean	Indicates whether the unique row count aggregate feature in Lightning Experience Reports is available in Salesforce (`true`) or not (`false`). Available in API version 47.0. Removed in API version 48.0.
`enableRequestPrioritySchdl`	boolean	Indicates whether priority-based dataflow request scheduling is available in Salesforce (`true`) or not (`false`). Available in API version 50.0 and later.
`enableS1AnalyticsEclairEnable`	boolean	Indicates whether EclairNG charts can be enabled for S1 Mobile Analytics in Salesforce (`true`) or not (`false`). Available in API version 48.0 and later.
`enableS3OutputConnector`	boolean	Indicates whether the S3 output data connector is available in Salesforce (`true`) or not (`false`). Available in API version 49.0 and later.

Field Name	Field Type	Description
enableSFXJoinedReports Enable	boolean	Indicates whether the Lightning Experience joined report feature can be enabled or disabled on the Setup page in Salesforce (true) or not (false). Available in API version 47.0 and later.
enableSalesforceOutput Connector	boolean	Indicates whether the Salesforce output data connector is available in Salesforce (true) or not (false). Available in API version 51.0 and later.
enableSecureImageSharing	boolean	Indicates whether secure image sharing and downloading is enabled in Salesforce (true) or not (false). Available in API version 50.0 and later.
enableSmartDataDiscovery	boolean	Indicates whether the org admin can enable Einstein Discovery in Salesforce (true) or not (false). Available in API version 49.0 and 50.0. Removed in API version 51.0.
enableSnowflakeOutput Connector	boolean	Indicates whether the Snowflake output data connector is available in Salesforce (true) or not (false). Available in API version 49.0 and later.
enableSummaryFilterOrg Pref	boolean	Indicates whether Lightning Experience Report summary filters are enabled (true) or not (false). Available in API version 60.0 and later.
enableSqlDataset	boolean	Indicates whether SQL datasets are available in Salesforce (true) or not (false). Available in API version 52.0. Removed in API version 53.0.
enableSqlLiveDataset	boolean	Indicates whether SQL live datasets are available in Salesforce (true) or not (false). Available in API version 51.0 and 52.0. Removed in API version 53.0.
enableTableauHyperOutput Connector	boolean	Indicates whether the Tableau hyper output data connector is available in Salesforce (true) or not (false). Available in API version 51.0 and later.
enableUseOldChartsLookAnd Feel	boolean	Indicates whether this org allows the old charts look and feel for Lightning Experience reports and dashboards (true) or not (false). Available in API version 47.0 and later.
enableWaveAssetsNewDate Version	boolean	Indicates whether the new date version for timezone support in Analytics assets is enabled in Salesforce (true) or not (false). Available in API version 51.0 and later.
enableWaveCustomFiscal	boolean	Indicates whether custom fiscal year is enabled for Analytics in Salesforce (true) or not (false). When enabled, custom fiscal year lets admins

Field Name	Field Type	Description
		import custom fiscal year definitions from Salesforce to Analytics. Available in API version 50.0 and later.
`enableWaveIndexMVDim`	boolean	Indicates whether multivalve dimension indexing is enabled in Salesforce (`true`) or not (`false`). Available in API version 50.0 and later.
`enableWaveIndexMVDimV2`	boolean	Indicates whether version 2 multivalve dimension indexing is enabled in Salesforce (`true`) or not (`false`). Available in API version 52.0 and later.
`enableWaveMulticurrency`	boolean	Indicates whether CRM Analytics multiple currencies is enabled (`true`) or not (`false`). Available in API version 56.0 and later. (Beta)
`enableWaveLwcDashboards`	boolean	Indicates whether embedded Analytics dashboards are rendered in Lightning Experience using a Lightning Web Component (`true`) or the legacy Aura Component (`false`). Available in API version 55.0 and 56.0. Removed in API version 57.0.
`enableWaveRecord Navigation`	boolean	Indicates whether browser tab navigation for record actions from Analytics is enabled in Salesforce (`true`) or not (`false`). Available in API version 48.0 and later.
`enableWaveReplication`	boolean	Indicates whether replication (extract) for Salesforce objects is enabled in Salesforce (`true`) instead of SFDC Digest (`false`). Available in API version 47.0 and later.
`enableWaveSharing Inheritance`	boolean	Indicates whether Analytics data can inherit sharing and security settings for their source object in Salesforce (`true`) or not (`false`). Available in API version 47.0 and later.
`enableWaveSqlCFIndexing`	boolean	Indicates whether indexing for custom fiscal dates in SQL queries is enabled in Salesforce (`true`) or not (`false`). Available in API version 58.0 and later.
`enableWaveSqlInQueryApi`	boolean	Indicates whether SQL is enabled for CRM Analytics in the Query API in Salesforce (`true`) or not (`false`). Available as Beta in API version 53.0. Removed for GA in API version 54.0.
`enableWaveTemplate`	boolean	Indicates whether Analytics templates are enabled for this org (`true`) or not (`false`). Removed in API version 51.0.
`enableWaveTrendedDataset Cleanup`	boolean	Indicates whether this org allows automatic deletion of inactive trended datasets (`true`) or not (`false`). Available in API version 47.0 and later.

Field Name	Field Type	Description
enableWriteToDataCloud	boolean	Indicates whether CRM Analytics recipe output to Data Cloud is enabled (true) or not (false). Available in API version 60.0 and later. (Beta)
etlOrchestrationPref	boolean	Indicates whether Data Prep recipe orchestration is enabled (true) or not (false). Available in API version 54.0 and later. (Beta)
inheritSharingForNonOppty Objects	boolean	Indicates whether medium visibility support for Analytics sharing inheritance for all Salesforce objects besides the Opportunity object is available in Salesforce (true) or not (false). Available in API version 48.0 and 49.0. Removed in API version 50.0.
inheritSharingForOppty Object	boolean	Indicates whether medium visibility support for Analytics sharing inheritance for the Opportunity object is available in Salesforce (true) or not (false). Available in API version 48.0 and 49.0. Removed in API version 50.0.
isDiscoveryOptimization Enabled	boolean	Indicates whether Einstein Discovery optimization is enabled (true) or not (false). Available in API version 58.0 and later. (Beta)
isHighVolumePushback Enabled	boolean	Indicates whether Einstein Discovery high volume push back is enabled in Salesforce (true) or not (false). Available in API version 58.0 and later.
maxHoursAppInProgress	integer	The maximum number of hours an embedded application can have the status InProgress before it's canceled. Available in API version 50.0 and later.
queryCachingOptOut	boolean	Indicates whether the option to cache query results is enabled (true) or not (false). Available in API version 59.0 and later.
recipeDirectDataPref	boolean	Indicates whether the option to enable Data Prep recipe direct data loading is available (true) or not (false). Available in API version 53.0 and later.
recipeFiscalPref	boolean	Indicates whether the option to enable Data Prep recipe custom fiscal settings is available (true) or not (false). Available in API version 53.0 and later.
recipePreCachingOptOut	boolean	Indicates whether the option to disable Data Prep recipe pre-step caching is available (true) or not (false). Available in API version 53.0 and later.
recipeStagedDataPref	boolean	Indicates whether staged data for Data Prep recipes is available in Salesforce (true) or not (false). Available in API version 53.0 and later.

Field Name	Field Type	Description
`replaceBlankMeasuresWithNulls`	boolean	Indicates whether null values are supported in measures (`true`) or not (`false`). If enabled (`true`), the implicit default value for blank measures is null. Available in API version 48.0 and later.
`setWaveIsYearEndFiscalYear`	boolean	Indicates whether the Analytics year end is the fiscal year end (`true`) or not (`false`). This preference is only applicable when `enableWaveCustomFiscal` is `true`. If `false`, the fiscal year end is the calendar year end. Available in API version 50.0 and later.
`sonicEnabled`	boolean	Indicates whether the Sonic feature is available in Salesforce (`true`) or not (`false`). Available in API version 51.0 and later.
`turnOnTimeZones`	boolean	Indicates whether the timezone feature is available in Salesforce (`true`) or not (`false`). Available in API version 48.0 and later.

Declarative Metadata Sample Definition

The following is an example of the Analytics.settings file:

```
<?xml version="1.0" encoding="UTF-8"?>
<AnalyticsSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <enableWaveTemplate>true</enableWaveTemplate>
    <enableInsights>true</enableInsights>
    <canAccessAnalyticsViaAPI>true</canAccessAnalyticsViaAPI>
</AnalyticsSettings>
```

Example Package Manifest

The following is an example package manifest used to deploy or retrieve the Analytics settings metadata:

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>Analytics</members>
        <name>Settings</name>
    </types>
    <version>47.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ApexSettings

Represents Apex-related org settings. This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

ApexSettings values are stored in the `Apex.settings` file in the `settings` directory of the corresponding package directory. The `.settings` files are different from other named components, because there's only one settings file for each settings component.

Version

ApexSettings components are available in API version 47.0 and later.

Fields

Field Name	Field Type	Description
`DefaultQueueableDelay`	integer	Indicates the admin-controlled minimum delay (in seconds) that applies to all enqueued jobs that were scheduled without a delay parameter. The minimum delay is one second and the maximum is 600 seconds. The default behavior, when the setting is omitted, is no delay in scheduling enqueued jobs.
`enableAggregateCodeCoverageOnly`	boolean	Indicates whether aggregate (not detailed) totals are tracked for Apex test coverage data (`true`) or not (`false`). The default value is `false`.
`enableApexAccessRightsPref`	boolean	Deprecated.
`enableApexApprovalLockUnlock`	boolean	Indicates whether approval process lock and unlock operations from Apex code are allowed (`true`) or not (`false`). The default value is `false`.
`enableApexCtrlImplicitWithSharingPref`	boolean	Indicates whether the Use `with sharing` for `@AuraEnabled` Apex Controllers with Implicit Sharing critical update is activated (`true`) or not (`false`). For more details, see the Winter '20 Release Notes.
`enableApexPropertyGetterPref`	boolean	Indicates whether the Enforce Access Modifiers on Apex Properties in Lightning Component Markup critical update is activated (`true`) or not (`false`). For more details, see the Winter '20 Release Notes.
`enableAuraApexCtrlAuthUserAccessCheckPref`	boolean	Indicates whether the Restrict Access to `@AuraEnabled` Apex Methods for Authenticated Users Based on User Profile critical

Field Name	Field Type	Description
		update is activated (`true`) or not (`false`). For more details, see the Winter '20 Release Notes.
`enableAuraApexCtrlGuestUserAccessCheckPref`	boolean	Indicates whether the Restrict Access to `@AuraEnabled` Apex Methods for Guest and Portal Users Based on User Profile critical update is activated (`true`) or not (`false`). For more details, see the Winter '20 Release Notes.
`enableCompileOnDeploy`	boolean	Indicates whether Apex code is automatically recompiled (`true`) or not (`false`). When set to `true`, code is recompiled before completing a metadata deployment, change set deployment, package installation, or package upgrade. For production orgs and full sandboxes, the default value is `true`. For all other orgs, the default value is `false`. Note: This setting can't be disabled in production orgs.
`enableDisableParallelApexTesting`	boolean	Indicates whether Apex tests are serially executed (`true`) or not (`false`). The default value is `false`. Note: Even when parallel testing is enabled by setting this value to `false`, tests that are run during deployments are always run serially.
`enableDebugLogsDuringDeployment`	boolean	Indicates whether debug logs are enabled during metadata deployment (`true`) or not (`false`), in conjunction with an active debug log trace flag. If this setting is `true` but the trace flag is not active, debug logs are not enabled during metadata deployment. The default value is `false`. Note: Enabling the debug log in the `DebuggingHeader` overrides this setting.
`enableGaplessTestAutoNum`	boolean	Indicates whether autonumbering gaps are prevented by Apex test executions not incrementing autonumber fields for non-test records (`true`) or not (`false`). The default value is `true`.
`enableMngdCtrlActionAccessPref`	boolean	Indicates whether the Disable Access to Non-global Apex Controller Methods in Managed Packages

Field Name	Field Type	Description
		critical update is activated (`true`) or not (`false`). For more details, see the Winter '20 Release Notes.
`enableNonCertifiedApexMdCrud`	boolean	Indicates whether Apex classes can access metadata, public or protected, through classes in the `Metadata` namespace (`true`) or not (`false`). The default value is `false`.
`enableRestrictCommunityExecAnon`	boolean	Indicates whether guest users are restricted from executing anonymous Apex. The restriction applies regardless of whether the `Author Apex` permission is set. The default value is `true`.
`enableSecureNoArgConstructorPref`	boolean	Not used. Critical update is automatically enforced. See Restrict Reflective Access to Non-Global Constructors in Packages (Critical Update).
`enableTestSetupSkipTestResults`	boolean	Indicates whether Apex test results are generated for @TestSetup methods (`false`) or not (`true`). Available in API version 61.0 and later.

Declarative Metadata Sample Definition

The following is an example of ApexSettings components.

```
<?xml version="1.0" encoding="UTF-8"?>
<ApexSettings xmlns="http://soap.sforce.com/2006/04/metadata">
   <enableDisableParallelApexTesting>true</enableDisableParallelApexTesting>
</ApexSettings>
```

The following is an example `package.xml` manifest that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>Apex</members>
        <name>Settings</name>
    </types>
    <version>47.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

AppAnalyticsSettings

Represents settings to retrieve AppExchange App Analytics usage data.

This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

AppAnalyticsSettings values are stored in the `AppAnalytics.settings` file in the `settings` folder. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

AppAnalyticsSettings is available in API versions 50.0 and later.

Fields

Field Name	Field Type	Description
`enableAppAnalyticsOptOut`	boolean	Indicates whether the collection of AppExchange App Analytics package usage data from this org is disabled (`true`) or not (`false`). Available in API version 60.0 and later. Default is `false`.
`enableSimulationMode`	boolean	Indicates whether AppExchange App Analytics simulation mode is enabled (`true`) or disabled (`false`). Available in API version 50.0 and later. Default is `false`.

Declarative Metadata Sample Definition

The following is an example of the AppAnalytics.settings file:

```
<?xml version="1.0" encoding="UTF-8"?>
<AppAnalyticsSettings xmlns="http://soap.sforce.com/2006/04/metadata">
 <enableSimulationMode>true</enableSimulationMode>
 <enableAppAnalyticsOptOut>false</enableAppAnalyticsOptOut>
</AppAnalyticsSettings>
```

Example Package Manifest

This example package manifest deploys or retrieves AppAnalytics settings metadata.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
 <types>
  <members>AppAnalytics</members>
  <name>Settings</name>
 </types>
 <version>50.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

AppExperienceSettings

Represents settings for the app experience.This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

AppExperienceSettings values are stored in the `.settings` file in the `settings` directory. The .settings files are different from other named components because there is only one settings file for each settings component.

Version

AppExperienceSettings components are available in API version 47.0 and later.

Fields

Field Name	Field Type	Description
`doesHideAllAppsInAppLauncher`	boolean	If set to `false` (default), all standard and custom apps show up on the App Launcher. If set to `true`, the admin must select which standard and custom apps to display on the App Launcher.

Declarative Metadata Sample Definition

The following is an example of an AppExperienceSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<AppExperienceSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <doesHideAllAppsInAppLauncher>false</doesHideAllAppsInAppLauncher>
</AppExperienceSettings>
```

Example Package Manifest

The following is an example package manifest used to deploy or retrieve the AppExperienceSettings metadata for an organization:

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>AppExperience</members>
        <name>Settings</name>
    </types>
<version>47.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

AssociationEngineSettings

Represents the record association builder settings for an org. This type extends the Metadata metadata type and inherits its `fullName` field.

Parent Type and Manifest Access

This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all the settings metadata types for the org are accessed using the "Settings" name. See Settings for more details.

File Suffix and Directory Location

`AssociationEngineSettings` values are stored in the `AssociationEngine.settings` file in the `settings` folder. The `.settings` files are different from other named components, because there is only one settings file for each settings component.

Version

AssociationEngineSettings components are available in API version 52.0 and later.

Special Access Rules

There are no additional access requirements that are specific to this type.

Fields

Field Name	Description
`enableAssociationEngine`	**Field Type** boolean **Description** Automatically associates new accounts with the user's current branch by creating branch unit customer records. The default value is `false`.

Declarative Metadata Sample Definition

The following is an example of an AssociationEngineSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<AssociationEngineSettings xmlns="http://soap.sforce.com/2006/04/metadata">
  <enableAssociationEngine>true</enableAssociationEngine>
</AssociationEngineSettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
  <types>
    <members>AssociationEngine</members>
    <name>Settings</name>
  </types>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

AutomatedContactsSettings

Represents an org's Einstein Automated Contacts settings. These settings let you find new contacts and opportunity contact roles. This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

AutomatedContactsSettings values are stored in the `AutomatedContacts.settings` file in the `settings` folder. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

AutomatedContactsSettings is available in API versions 48.0 and later.

Fields

Field Name	Field Type	Description
`enableAddContactAutomatically`	boolean	Indicates whether new contacts are automatically added from external email accounts (such as Microsoft and Google) to Salesforce (`true`) or not (`false`). The default value is `false`. `enableAddContactWithSuggestion` must be `true` to use this setting. Note: When this feature is enabled, users do not see new contacts as suggestions. The contacts are added automatically.

Field Name	Field Type	Description
enableAddContactRoleAutomatically	boolean	Indicates whether opportunity contact roles from external accounts are automatically added to Salesforce (true) or not (false). The default value is false. Note: When this feature is enabled, users do not see new contact roles as suggestions. The contact roles are added automatically.
enableAddContactRoleWithSuggestion	boolean	Indicates whether opportunity contact roles from external accounts are suggested as new Salesforce opportunity contact roles (true) or not (false). The default value is true.
enableAddContactWithSuggestion	boolean	Indicates whether new contacts from external accounts (such as Microsoft and Google) are suggested as new Salesforce contacts (true) or not (false). The default value is true. enableAddContactRoleWithSuggestion must be true to use this setting.

Declarative Metadata Sample Definition

The following is an example of the AutomatedContactsSettings.settings file:

```
<?xml version="1.0" encoding="UTF-8"?>
<AutomatedContactsSettings xmlns="http://soap.sforce.com/2006/04/metadata">
 <enableAddContactAutomatically>true</enableAddContactAutomatically>
 <enableAddContactRoleAutomatically>true</enableAddContactRoleAutomatically>
 <enableAddContactRoleWithSuggestion>true</enableAddContactRoleWithSuggestion>
 <enableAddContactWithSuggestion>true</enableAddContactWithSuggestion>

</AutomatedContactsSettings>
```

Example Package Manifest

The following is an example package manifest used to deploy or retrieve the AutomatedContacts settings metadata:

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
 <types>
  <members>AutomatedContactsSettings</members>
  <name>Settings</name>
 </types>
 <version>29.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character * (asterisk) in the package.xml manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

BotSettings

Represents an organization's Einstein Bot settings, such as whether or not Einstein Bots is enabled. This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

Bot components have the suffix `.bot` and are stored in the `bot` folder.

Version

Bot components are available in API version 46.0 and later.

Fields

Field Name	Field Type	Description
`enableBots`	boolean	Indicates whether Einstein Bots is enabled (`true`) or not (`false`).

Declarative Metadata Sample Definition

The following is an example of a BotSetting. This example has been trimmed to make it easier to read.

```
<?xml version="1.0" encoding="UTF-8"?>
<BotSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <enableBots>true</enableBots>
</BotSettings>
```

The following is an example `package.xml` manifest that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>Bot</members>
        <name>Settings</name>
    </types>
    <version>46.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

BranchManagementSettings

Represents the branch management settings for an org. This type extends the Metadata metadata type and inherits its `fullName` field.

Parent Type and Manifest Access

This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all the settings metadata types for the org are accessed using the "Settings" name. See Settings for more details.

File Suffix and Directory Location

`BranchManagementSettings` values are stored in the `BranchManagement.settings` file in the `settings` folder. The `.settings` files are different from other named components, because there is only one settings file for each settings component.

Version

BranchManagementSettings components are available in API version 51.0 and later.

Special Access Rules

There are no additional access requirements that are specific to this type.

Fields

Field Name	Description
`associateAccountWithBranch`	**Field Type** boolean **Description** Automatically associates new accounts with the user's current branch by creating branch unit customer records. The default value is `false`.

Declarative Metadata Sample Definition

The following is an example of a BranchManagementSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<BranchManagementSettings xmlns="http://soap.sforce.com/2006/04/metadata">
  <associateAccountWithBranch>true</associateAccountWithBranch>
</BranchManagementSettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>BranchManagement</members>
        <name>Settings</name>
    </types>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

BusinessHoursSettings

Represents the metadata used to manage settings for business hours and holidays in entitlements, entitlement templates, campaigns, and cases. This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

Business hours and holidays settings are stored in a single file named `businessHours.settings` in the `settings` directory. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

BusinessHoursSettings is available in API version 29.0 and later.

Fields

Field Name	Field Type	Description
`businessHours`	BusinessHoursEntry[]	Represents the application of business hours to entitlements, entitlement templates, campaigns, and cases.
`holidays`	Holidays[]	Represents a holiday and its usage in `businessHours`.

BusinessHoursEntry

Represents the application of business hours to entitlements, entitlement templates, campaigns, and cases.

Field Name	Field Type	Description
`timeZoneId`	string	The time zone for the time that defines business hours.
`name`	string	Name of the business hours. This name should be unique.
`active`	string	Indicates whether the business hours are active.
`default`	string	Indicates whether the business hours are used as the default business hours.
`mondayStartTime`	string	Start time for the business hours on Monday. Uses the format `HH:mm:ss.SSSZ`.

Field Name	Field Type	Description
mondayEndTime	string	End time for the business hours on Monday. Uses the format HH:mm:ss.SSSZ. The value 00:00:00.000Z specifies midnight on Monday.
tuesdayStartTime	string	Start time for the business hours on Tuesday. Uses the format HH:mm:ss.SSSZ.
tuesdayEndTime	string	End time for the business hours on Tuesday. Uses the format HH:mm:ss.SSSZ. The value 00:00:00.000Z specifies midnight on Tuesday.
wednesdayStartTime	string	Start time for the business hours on Wednesday. Uses the format HH:mm:ss.SSSZ.
wednesdayEndTime	string	End time for the business hours on Wednesday. Uses the format HH:mm:ss.SSSZ. The value 00:00:00.000Z specifies midnight on Wednesday.
thursdayStartTime	string	Start time for the business hours on Thursday. Uses the format HH:mm:ss.SSSZ.
thursdayEndTime	string	End time for the business hours on Thursday. Uses the format HH:mm:ss.SSSZ. The value 00:00:00.000Z specifies midnight on Thursday.
fridayStartTime	string	Start time for the business hours on Friday. Uses the format HH:mm:ss.SSSZ.
fridayEndTime	string	End time for the business hours on Friday. Uses the format HH:mm:ss.SSSZ. The value 00:00:00.000Z specifies midnight on Friday.
saturdayStartTime	string	Start time for the business hours on Saturday. Uses the format HH:mm:ss.SSSZ.
saturdayEndTime	string	End time for the business hours on Saturday. Uses the format HH:mm:ss.SSSZ. The value 00:00:00.000Z specifies midnight on Saturday.
sundayStartTime	string	Start time for the business hours on Sunday. Uses the format HH:mm:ss.SSSZ.
sundayEndTime	string	End time for the business hours on Sunday. Uses the format HH:mm:ss.SSSZ. The value 00:00:00.000Z specifies midnight on Sunday.

Holidays

Represents a holiday and its usage in businessHours.

Field Name	Field Type	Description
`name`	string	Name of the holiday. This name does not have to be unique.
`description`	string	The description of the holiday.
`isRecurring`	string	Indicates whether the holiday is recurring.
`activityDate`	string	The date of the holiday. Use for non-recurring holidays. Uses the format `HH:mm:ss.SSSZ`.
`recurrenceStartDate`	string	The date the holiday starts recurring. Uses the format `yyyy-mm-dd`.
`recurrenceEndDate`	string	The date the holiday stops recurring. Uses the format `yyyy-mm-dd`. Optional.
`startTime`	string	The start time on the date of the holiday. Uses the format `HH:mm:ss.SSSZ`. `startTime` and `endTime` must be both null or both not null. If they are both null, indicates the whole day.
`endTime`	string	The end time on the date of the holiday. Uses the format `HH:mm:ss.SSSZ`. `startTime` and `endTime` must be both null or both not null. If they are both null, indicates the whole day.
`recurrenceType`	string	The recurrence type of the holiday. Valid values are: RecursDaily, RecursEveryWeekday, RecursMonthly, RecursMonthlyNth, RecursWeekly, RecursYearly, RecursYealyNth.
`recurrenceInterval`	string	The interval of weeks, months, or years the holiday recurs.
`recurrenceDayOfWeek`	string	The day of week the holiday recurs. Valid values: Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday.
`recurrenceDayOfMonth`	string	The day of month the holiday recurs. Valid values: integers 1-31.
`recurrenceInstance`	string	Valid values: First, Second, Third, Fourth, Last. Only used for recurrenceType RecursMonthlyNth and RecursYearlyNth. For example, if the recurenceInstance value is First, the holiday recurs on the first Monday of the month every 3 months.
`recurrenceMonthOfYear`	string	Valid values: January, February, March, April, May, June, July, August, September, October, November, December.
`businessHours`	string	The name of the business hours setting that applies to this holiday.

Declarative Metadata Sample Definition

The following is an example `businesshours.settings` metadata file:

```
<?xml version="1.0" encoding="UTF-8"?>
<BusinessHoursSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <businessHours>
        <active>true</active>
        <default>true</default>
        <fridayEndTime>00:00:00.000Z</fridayEndTime>
```

```
        <fridayStartTime>00:00:00.000Z</fridayStartTime>
        <mondayEndTime>00:00:00.000Z</mondayEndTime>
        <mondayStartTime>00:00:00.000Z</mondayStartTime>
        <name>Default</name>
        <saturdayEndTime>00:00:00.000Z</saturdayEndTime>
        <saturdayStartTime>00:00:00.000Z</saturdayStartTime>
        <sundayEndTime>00:00:00.000Z</sundayEndTime>
        <sundayStartTime>00:00:00.000Z</sundayStartTime>
        <thursdayEndTime>00:00:00.000Z</thursdayEndTime>
        <thursdayStartTime>00:00:00.000Z</thursdayStartTime>
        <timeZoneId>America/Los_Angeles</timeZoneId>
        <tuesdayEndTime>00:00:00.000Z</tuesdayEndTime>
        <tuesdayStartTime>00:00:00.000Z</tuesdayStartTime>
        <wednesdayEndTime>00:00:00.000Z</wednesdayEndTime>
        <wednesdayStartTime>00:00:00.000Z</wednesdayStartTime>
    </businessHours>
    <businessHours>
        <active>true</active>
        <default>false</default>
        <fridayEndTime>00:00:00.000Z</fridayEndTime>
        <fridayStartTime>00:00:00.000Z</fridayStartTime>
        <mondayEndTime>15:00:00.000Z</mondayEndTime>
        <mondayStartTime>09:00:00.000Z</mondayStartTime>
        <name>bh1</name>
        <saturdayEndTime>00:00:00.000Z</saturdayEndTime>
        <saturdayStartTime>00:00:00.000Z</saturdayStartTime>
        <sundayEndTime>00:00:00.000Z</sundayEndTime>
        <sundayStartTime>00:00:00.000Z</sundayStartTime>
        <thursdayEndTime>17:00:00.000Z</thursdayEndTime>
        <thursdayStartTime>10:50:00.000Z</thursdayStartTime>
        <timeZoneId>America/Los_Angeles</timeZoneId>
        <tuesdayEndTime>13:00:00.000Z</tuesdayEndTime>
        <tuesdayStartTime>09:00:00.000Z</tuesdayStartTime>
        <wednesdayEndTime>15:00:00.000Z</wednesdayEndTime>
        <wednesdayStartTime>09:00:00.000Z</wednesdayStartTime>
    </businessHours>
    <holidays>
        <activityDate>2013-09-02</activityDate>
        <businessHours>Default</businessHours>
        <businessHours>bh1</businessHours>
        <isRecurring>false</isRecurring>
        <name>Labor Day</name>
    </holidays>
    <holidays>
        <businessHours>bh1</businessHours>
        <isRecurring>true</isRecurring>
        <name>Christmas</name>
        <recurrenceDayOfMonth>25</recurrenceDayOfMonth>
        <recurrenceMonthOfYear>December</recurrenceMonthOfYear>
        <recurrenceStartDate>2013-12-25</recurrenceStartDate>
        <recurrenceType>RecursYearly</recurrenceType>
    </holidays>
</BusinessHoursSettings>
```

The following is an example `package.xml` manifest that references the BusinessHoursSettings definitions:

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>BusinessHours</members>
        <name>Settings</name>
    </types>
    <version>29.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

CampaignSettings

Represents an org's Campaign Influence, Einstein Attribution, Einstein Key Accounts, and campaign member settings. These features help you understand how your campaigns and accounts are affecting your opportunity pipeline.

This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

CampaignSettings values are stored in the `Campaign.settings` file in the `settings` folder. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

CampaignSettings is available in API versions 48.0 and later.

Fields

Field Name	Field Type	Description
`aiAttributionTimeframe`	int	Indicates the number of months between the opportunity creation date and an engagement activity, during which Einstein scans for influential campaigns. The value must be a multiple of three, up to 24. Available in API version 49.0 and later. This field supports Einstein Attribution.
`enableAIAttribution`	boolean	Indicates whether the Einstein Attribution feature is enabled (`true`) or not (`false`). The default value is `false`. Available in API version 49.0 and later.

Field Name	Field Type	Description
enableAccountsAsCM	boolean	Indicates whether accounts can be used as campaign members (true) or not (false). The default value is false. Available in API version 51.0 and later.
enableAutoCampInfluenceDisabled	boolean	Indicates whether Salesforce creates Campaign Influence information (true) or not (false). enableCampaignInfluence2 must be false to use this setting. The default value is false.
enableB2bmaCampaignInfluence2	boolean	Indicates whether your org can access campaign influence models from other systems, such as Pardot (true) or not (false). enableCampaignInfluence2 must be true to use this setting. The default value is false.
enableCampaignHistoryTrackEnabled	boolean	This read-only field is reserved for system use.
enableCampaignInfluence2	boolean	Indicates whether Customizable Campaign Influence is enabled (true) or not (false). When true, Campaign Influence 1.0 is hidden from users and is no longer active. The default value is true.
enableCampaignMemberTWCF	boolean	This read-only field is reserved for system use.
enableEKAI	boolean	Indicates whether Einstein Key Accounts Identification is enabled (true) or not (false). The default value is false. Available in API version 53.0 and later.
enableOpportunityInfluence		Indicates whether Opportunity Influence campaign attribution is enabled (true) or not (false). When enabled, Opportunity Influence connects opportunity revenue to specific campaigns. The default value is false. Available with Marketing Cloud Growth and Advanced editions in API version 63.0 and higher.
enableSuppressNoValueCI2	boolean	This read-only field is reserved for system use.

Declarative Metadata Sample Definition

The following is an example of the Campaign.settings file:

```
<?xml version="1.0" encoding="UTF-8"?>
<CampaignSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <enableCampaignInfluence2>true</enableCampaignInfluence2>
    <enableSuppressNoValueCI2>true</enableSuppressNoValueCI2>
    <enableCampaignHistoryTrackEnabled>true</enableCampaignHistoryTrackEnabled>
    <enableAutoCampInfluenceDisabled>true</enableAutoCampInfluenceDisabled>
    <enableCampaignMemberTWCF>true</enableCampaignMemberTWCF>
    <enableB2bmaCampaignInfluence2>true</enableB2bmaCampaignInfluence2>
    <enableAccountsAsCM>true</enableAccountsAsCM>
    <enableAIAttribution>true</enableAIAttribution>
```

```
    <aiAttributionTimeframe>9</aiAttributionTimeframe>
    <enableEKAI>true</enableEKAI>
</CampaignSettings>
```

Example Package Manifest

The following is an example package manifest used to deploy or retrieve the Campaign settings metadata:

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
 <types>
  <members>Campaign</members>
  <name>Settings</name>
 </types>
 <version>29.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

CaseSettings

Represents an organization's case settings, such as the default case owner, which case-related features are enabled, and which Classic email templates are used for various case activities. This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

CaseSettings on page 1856 values are stored in the `Case.settings` file in the `settings` directory. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

CaseSettings on page 1856 is available in API version 27.0 and later.

Fields

Field Name	Field Type	Description
`caseAssignNotificationTemplate`	string	Specifies the email template used for case assignment notifications. The format must be `folderName/emailTemplateName`. Lightning email templates aren't packageable. We recommend using a Classic email template.

Field Name	Field Type	Description
caseAutoProcUser	boolean	Indicates whether to create an automated response record after a customer's initial email (true) or not (false).
caseCloseNotificationTemplate	string	Specifies the email template used for case close notifications. The format must be folderName/emailTemplateName. Lightning email templates aren't packageable. We recommend using a Classic email template.
caseCommentNotificationTemplate	string	Specifies the email template used for case comment notifications. The format must be folderName/emailTemplateName. Lightning email templates aren't packageable. We recommend using a Classic email template.
caseCreateNotificationTemplate	string	Specifies the email template used for case create notifications. The format must be folderName/emailTemplateName. Lightning email templates aren't packageable. We recommend using a Classic email template.
caseFeedItemSettings	FeedItemSettings[]	Specifies the settings for feed items in feed-based case page layouts. This field is available in API version 32.0 and later.
caseFeedReadUnreadLtng	boolean	Indicates whether unread feed items are shown in bold in Lightning Experience (true) or not (false).
caseMergeInLightning	boolean	Indicates whether Case Merge is enabled in Lightning Experience (true) or not (false).
closeCaseThroughStatusChange	boolean	Indicates whether Closed is included in the Case Status field on case edit pages (true) or not (false).
defaultCaseFeedLayoutOn	boolean	Indicates whether the default Case Feed layout is used in the org (true) or not (false).
defaultCaseOwner	string	Specifies the default owner of a case when assignment rules fail to locate an owner.
defaultCaseOwnerType	string	Specifies whether the default case owner is a user or a queue.
defaultCaseUser	string	Specifies the user listed in the Case History related list for automated case changes from: • Assignment rules • Escalation rules • On-Demand Email-to-Case • Cases logged in the Self-Service portal

Field Name	Field Type	Description
		Lightning email templates aren't packageable. We recommend using a Classic email template.
`emailActionDefaultsHandlerClass`	string	Use this Apex class name to provide default values for the email action.
`emailToCase`	EmailToCaseSettings	The organization's Email-to-Case settings.
`enableCaseFeed`	boolean	Indicates whether Case Feed is enabled (`true`) or not (`false`).
`enableCollapseEmailThread`	boolean	Indicates whether earlier messages in an email thread are removed from email feed items (`true`) or not (`false`). Available in API version 47.0 and later.
`enableDraftEmails`	boolean	Indicates whether draft emails are enabled (`true`) or not (`false`). Enabling email drafts requires that Case Feed and Email-to-Case are also enabled.
`enableEarlyEscalationRuleTriggers`	boolean	Indicates whether early triggers on escalation rules are enabled (`true`) or not (`false`).
`enableEmailActionDefaultsHandler`	boolean	Indicates whether the Email Action Default Handler setting is enabled (`true`) or not (`false`). Use this setting to select an Apex class to load a default template or to specify the default target fields for the email action.
`enableEmailContactOnCasePost`	boolean	If `true`, the case contact is notified by email when someone makes an externally visible post on a case in an Experience Builder site.
`enableEscalateQfiToCaseInternal`	boolean	If `true`, moderators can create cases from Question feed items in Chatter in your organization.
`enableEscalateQfiToCaseNetworks`	boolean	If `true`, moderators can create cases from Question feed items in Chatter in all Experience Builder sites where Chatter Questions is enabled.
`enableExtNetworksCaseFeedEnabled`	boolean	If `true`, site members can see case-related emails, comments, and updates in the case feed.
`enableMultiLangSolnSrchCSS`	boolean	Indicates whether multilingual searching for Solutions in self-service portals is enabled (`true`) or not (`false`).
`enableMultiLangSolnSrchPKB`	boolean	Indicates whether multilingual searching for public Solutions is enabled (`true`) or not (`false`).
`enableMultiLangSolution`	boolean	Indicates whether multilingual Solutions are enabled (`true`) or not (`false`).

Field Name	Field Type	Description
enableNewEmailDefaultTemplate	boolean	Indicates whether default email templates are enabled (true) or not (false). Default email templates are available only if draft emails are enabled. Lightning email templates aren't packageable. We recommend using a Classic email template.
enableSolutionCategory	boolean	Indicates whether browsing for Solutions is enabled (true) or not (false).
enableSolutionInlineCategory	boolean	Indicates whether using inline Solutions category breadcrumbs is enabled (true) or not (false).
enableSolutionShortSummary	boolean	Indicates whether Solutions summaries are enabled (true) or not (false).
enableSuggestedArticlesApplication	boolean	Indicates whether the Suggested Articles list appears on case pages. (true) or not (false). Is only valid if enableSuggestedSolutions=false.
enableSuggestedArticlesCustomerPortal	boolean	Indicates whether the Suggested Articles list appears on customer portal pages (true) or not (false). Is only valid if enableSuggestedSolutions=false.
enableSuggestedArticlesPartnerPortal	boolean	Indicates whether the Suggested Articles list appears on partner portal pages (true) or not (false). Is only valid if enableSuggestedSolutions=false.
enableSuggestedSolutions	boolean	Indicates whether the View Suggested Solutions or Find Articles button appears on case detail pages (true) or not (false). Is only valid if enableSuggestedArticlesApplication, enableSuggestedArticlesCustomerPortal, and enableSuggestedArticlesPartnerPortal=false.
escalateCaseBefore	boolean	Indicates whether early triggers are enabled to escalate a case (true) or not (false).
genericMessageEnabled	boolean	Indicates whether generic messages are enabled (true) or not (false).
keepCaseMergeRecords	boolean	If true, duplicate cases aren't deleted after a case merge.
keepRecordTypeOnAssignmentRule	boolean	When applying assignment rules to manually created records, indicates whether to keep the existing record type (true) or to override the existing record type with the assignee's default record type (false).
newEmailDefaultTemplateClass	string	Specifies the Apex class that defines the default email template for new email messages in Case Feed. This field appears only when enableNewEmailDefaultTemplate=true.Lightning

Field Name	Field Type	Description
		email templates aren't packageable. We recommend using a Classic email template.
`notifyContactOnCaseComment`	boolean	Indicates whether contacts who aren't members of your Self-Service portal can be notified when a new comment is added to a case (`true`) or not (`false`).
`notifyDefaultCaseOwner`	boolean	Indicates whether the default case owner is notified when assigned a new case (`true`) or not (`false`).
`notifyOwnerOnCaseComment`	boolean	Indicates whether the case owner is notified when a comment is added to a case (`true`) or not (`false`).
`notifyOwnerOnCaseOwnerChange`	boolean	Indicates whether the `Send Notification Email` checkbox on cases is automatically selected when users change a case owner to another user (`true`).
`predictiveSupportEnabled`	boolean	Indicates whether predictive support is enabled (`true`) or not (`false`).
`showEmailAttachmentsInCase AttachmentsRL`	boolean	Indicates whether the case Attachments related list shows email attachments. If `true`, the page displays an email icon next to each attachment from an email in the Attachments related list for cases. The related list's list view also includes a Source column that identifies the attachment's origin. If `false`, email attachments aren't displayed in the Attachments related list for cases. This field is available in API version 40.0 and later.
`showFewerCloseActions`	boolean	Indicates whether the **Save & Close** button on case edit pages and the **Cls** link on Cases related lists are hidden (`true`) or shown (`false`).
`systemUserEmail`	string	Specifies the email address used when the default case user is the system user.
`useSystemEmailAddress`	boolean	Indicates whom case comment, case attachment, and case assignment email notifications appear to be sent from. Use `true` to show notifications as sent from a system address. Use `false` to show notifications as sent from the user or contact who is updating the case.
`useSystemUserAsDefaultCaseUser`	boolean	Indicates whether the system user is used as the automated case user (`true`) or not (`false`). If false, then you must specify a value for the `defaultCaseUser` field.
`visibleInCssCheckbox`	boolean	Sets the default visibility of a case as indicated by the Visible in CSS option on the case edit page. If `false`, the case is visible in CSS by default. If `true`, CSS visibility is off.
`webToCase`	WebToCaseSettings	The organization's Web-to-Case settings.

EmailToCaseSettings

Represents an organization's Email-to-Case settings.

Fields

Field Name	Field Type	Description
enableEmailToCase	boolean	Indicates whether Email-to-Case is enabled (true) or not (false). **Note:** After Email-to-Case is enabled, it can't be disabled.
enableE2CAttachmentAsFile	boolean	Indicates whether to save attachments sent using Email-to-Case as Salesforce Files (true) or not (false).
enableE2CDeduplicateAttachments	boolean	When Email-to-Case receives an inbound email to thread to an existing case, attachments in the email that already exist on the case are not saved as new records and are instead linked to the new email (true).
enableE2CExternalServer	boolean	Indicates whether emails can be sent via an external service such as Gmail or Outlook, rather than the Salesforce email service (true). External outbound email services are available in Lightning Experience only.
enableE2CSourceTracking	boolean	Indicates whether Set Case Source to Email is enabled (true) or not (false). After you enable this setting, the **Case Source** field is updated to Email for all cases that originate from Email-to-Case. Associated emails are marked as Read when the agent opens the case.
enableHtmlEmail	boolean	Indicates whether HTML email is enabled (true) or not (false).
enableNewtoReadTriggers	boolean	Indicates whether the email status change invokes triggers when emails open in Case feeds or from Email Message records in Lightning (true) or not (false).
enableOnDemandEmailToCase	boolean	Indicates whether On-Demand Email-to-Case is enabled (true) or not (false).
enableThreadIDInBody	boolean	Indicates whether the Thread ID for a case is inserted in the body of an email (true) or not (false). This is applicable only to orgs that do not use Lightning Threading.
enableThreadIDInSubject	boolean	Indicates whether the Thread ID for a case is inserted in the subject line of an email (true) or not (false). This is applicable only to orgs that do not use Lightning Threading.
enableThreadTokenInBody	boolean	Indicates whether a threading token is appended in the email body when agents send an email from a Lightning email composer (true) or not (false). This is applicable only to orgs using Lightning Threading.

Field Name	Field Type	Description
enableThreadTokenInSubject	boolean	Indicates whether a threading token is appended in the email subject when agents send an email from a Lightning email composer (true) or not (false). This is applicable only to orgs using Lightning Threading.
notifyOwnerOnNewCaseEmail	boolean	Indicates whether the owner of a case receives a notification when a new email related to the case is received (true) or not (false).
overEmailLimitAction	EmailToCaseOnFailureActionType (enumeration of type string)	Specifies what happens to email messages that are received after an organization exceeds its daily Email-to-Case limits. Valid values are: • Bounce • Discard • Requeue
preQuoteSignature	boolean	Indicates whether the user signature is inserted after the reply but before the email thread in an outbound email (true) or at the end of the email (false).
routingAddresses	EmailToCaseRoutingAddress[]	The organization's Email-to-Case routing addresses and their attributes. Removing an address from this list deletes it from the target org.
replyWithNewContentOnly	boolean	Indicates whether previous thread content in excluded from replies, to reduce the size of outgoing emails (true) or not (false).
showGeneratedEmailCheckbox	boolean	Indicates whether senders are required to confirm that they reviewed emails written by Einstein (true) or not (false).
unauthorizedSenderAction	EmailToCaseOnFailureActionType (enumeration of type string)	Specifies what happens to email messages received from invalid senders. Valid values are: • Bounce • Discard
useEmailHeadersForThreading	boolean	Indicates whether metadata from incoming emails is used to match replies with cases if token-based threading doesn't produce a match (true) or not (false).

EmailToCaseRoutingAddress

Represents an organization's Email-to-Case routing address.

Fields

Field Name	Field Type	Description
`addressType`	EmailToCaseRoutingAddressType (enumeration of type string)	Specifies the type of Email-to-Case routing address. Valid values are: • `EmailToCase`—A routing address used with Email-to-Case or On-Demand Email-to-Case. • `Outlook`—A routing address used with Salesforce for Outlook to create cases from Outlook. Requires that On-Demand Email-to-Case is enabled.
`authorizedSenders`	string	Specifies the email addresses or domains from which On-Demand Email-to-Case can receive email. Include multiple entries in a comma-separated list.
`caseOrigin`	string	Specifies the default case origin for cases created through this routing address.
`caseOwner`	string	Specifies the default owner of cases created through this routing address. The case owner can be either a user or a queue. Specify the case owner using a Salesforce username. Specifying a case owner here in the routing address sets a value of `defaultCaseOwner` in CaseSettings.
`caseOwnerType`	string	Specifies whether the default case owner is a user or a queue.
`casePriority`	string	Specifies the default case priority for cases created through this routing address.
`createTask`	boolean	Indicates whether a task is automatically assigned to the case owner when a case is created through an email (`true`) or not (`false`).
`emailAddress`	string	Specifies the email address used to route email messages that are submitted as cases.
`emailServicesAddress`	string	Specifies the Salesforce-generated routing address used for setting up Email-to-Case forwarding. This field value is read-only and can't be modified.
`fallbackQueue`	string	Defines which queue to use when emails can't be routed with the specified Omni-Channel flow. This queue must use Case as the service channel object. Available in API version 56.0 and later.
`isPermsetControlled`	boolean	Indicates whether users' access to the email routing address is controlled by a permission set. If `true`, only users with access via a permission set can use the routing address to send emails.

Field Name	Field Type	Description
`isVerified`	boolean	Indicates whether the customer has verified the routing address (typically by clicking a confirmation email). This field value is read-only and can't be modified.
`routingFlow`	string	Specifies the name of an Omni-Channel flow that routes cases generated in Email-to-Case. Available in API version 56.0 and later.
`routingName`	string	Specifies the name of the Email-to-Case routing address.
`saveEmailHeaders`	boolean	Indicates whether email routing and envelope information are saved (`true`) or not (`false`).
`taskStatus`	string	Specifies the default status on tasks automatically assigned to the case owner when email is submitted as a case. Only applies if `createTask` is set to `true`.

FeedItemSettings

Represents an organization's feed item settings. Available in API version 32.0 and later.

Field Name	Field Type	Description
`characterLimit`	int	Specifies the maximum number of characters displayed for each feed item.
`collapseThread`	boolean	Removed. Indicates whether earlier messages in an email thread are removed from email feed items (`true`) or not (`false`). Available in API versions 27.0 to 46.0.
`displayFormat`	FeedItemDisplayFormat (enumeration of type string)	Indicates how email feed items are displayed. Valid values are: • `Default`—Blank lines in email feed items are displayed. • `HideBlankLines`—Blank lines in email feed items aren't displayed.
`feedItemType`	FeedItemType (enumeration of type string)	The type of feed item to which the settings apply. For `FeedItemSettings`, the only valid `feedItemType` value is `EmailMessageEvent`.

WebToCaseSettings

Represents an organization's Web-to-Case settings.

Fields

Field Name	Field Type	Description
caseOrigin	string	Specifies the default case origin for cases created through this web form. Applies only if enableWebToCase is set to true.
defaultResponseTemplate	string	Specifies the default template used for email responses to cases that are submitted through a Self-Service portal. Applies only if enableWebToCase is set to true. Lightning email templates aren't packageable. We recommend using a Classic email template.
enableWebToCase	boolean	Indicates whether Web-to-Case is enabled (true) or not (false).

Declarative Metadata Sample Definition

This code sample is an example of a case settings file.

```
<?xml version="1.0" encoding="UTF-8"?>
<CaseSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <caseAssignNotificationTemplate>
        unfiled$public/SupportCaseAssignmentNotification
    </caseAssignNotificationTemplate>
    <caseCloseNotificationTemplate>
        unfiled$public/SupportCaseCloseNotification
    </caseCloseNotificationTemplate>
    <caseCommentNotificationTemplate>
        unfiled$public/SupportCaseCommentNotification
    </caseCommentNotificationTemplate>
    <caseCreateNotificationTemplate>
        unfiled$public/SupportCaseCreateNotification
    </caseCreateNotificationTemplate>
    <closeCaseThroughStatusChange>true</closeCaseThroughStatusChange>
    <defaultCaseOwner>admin@acme.com</defaultCaseOwner>
    <defaultCaseOwnerType>User</defaultCaseOwnerType>
    <defaultCaseUser>admin@acme.com</defaultCaseUser>
    <emailToCase>
        <enableEmailToCase>true</enableEmailToCase>
        <enableHtmlEmail>false</enableHtmlEmail>
        <enableOnDemandEmailToCase>true</enableOnDemandEmailToCase>
        <enableThreadIDInBody>true</enableThreadIDInBody>
        <enableThreadIDInSubject>true</enableThreadIDInSubject>
        <notifyOwnerOnNewCaseEmail>false</notifyOwnerOnNewCaseEmail>
        <overEmailLimitAction>Bounce</overEmailLimitAction>
        <preQuoteSignature>true</preQuoteSignature>
        <routingAddresses>
            <addressType>EmailToCase</addressType>
            <authorizedSenders>user@acme.com</authorizedSenders>
            <caseOrigin>Email</caseOrigin>
            <casePriority>Medium</casePriority>
            <createTask>true</createTask>
```

```
            <emailAddress>support@acme.com</emailAddress>
            <routingName>EmailToCaseRoutingAddress1</routingName>
            <saveEmailHeaders>true</saveEmailHeaders>
            <taskStatus>Not Started</taskStatus>
        </routingAddresses>
        <routingAddresses>
            <addressType>Outlook</addressType>
            <authorizedSenders>user@acme.com</authorizedSenders>
            <caseOrigin>Email</caseOrigin>
            <caseOwner>admin@acme.com</caseOwner>
            <caseOwnerType>User</caseOwnerType>
            <casePriority>High</casePriority>
            <routingName>OutlookRoutingAddress1</routingName>
        </routingAddresses>
        <unauthorizedSenderAction>Discard</unauthorizedSenderAction>
    </emailToCase>
    <enableCaseFeed>true</enableCaseFeed>
    <enableDraftEmails>true</enableDraftEmails>
    <enableEarlyEscalationRuleTriggers>true</enableEarlyEscalationRuleTriggers>
    <enableNewEmailDefaultTemplate>true</enableNewEmailDefaultTemplate>
    <enableSuggestedArticlesApplication>true</enableSuggestedArticlesApplication>
    <enableSuggestedArticlesCustomerPortal>true</enableSuggestedArticlesCustomerPortal>
    <enableSuggestedArticlesPartnerPortal>false</enableSuggestedArticlesPartnerPortal>
    <enableSuggestedSolutions>false</enableSuggestedSolutions>
    <keepRecordTypeOnAssignmentRule>true</keepRecordTypeOnAssignmentRule>
    <newEmailDefaultTemplateClass>CaseTemplateController</newEmailDefaultTemplateClass>
    <notifyContactOnCaseComment>true</notifyContactOnCaseComment>
    <notifyDefaultCaseOwner>true</notifyDefaultCaseOwner>
    <notifyOwnerOnCaseComment>true</notifyOwnerOnCaseComment>
    <notifyOwnerOnCaseOwnerChange>false</notifyOwnerOnCaseOwnerChange>
    <showFewerCloseActions>false</showFewerCloseActions>
    <useSystemEmailAddress>true</useSystemEmailAddress>
    <webToCase>
        <caseOrigin>Web</caseOrigin>
       <defaultResponseTemplate>unfiled$public/SupportCaseResponse</defaultResponseTemplate>

        <enableWebToCase>true</enableWebToCase>
    </webToCase>
</CaseSettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>Case</members>
        <name>Settings</name>
    </types>
    <version>47.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ChatterAnswersSettings

Represents the metadata used to manage settings for Chatter Answers.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

Chatter Answers settings are stored in a single file named `ChatterAnswers.settings` in the `settings` directory. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

ChatterAnswersSettings is available in API version 27.0 and later.

Fields

Field Name	Field Type	Description
`emailFollowersOnBestAnswer`	boolean	Indicates whether users are notified when a best answer is selected for a question that they're following (`true`) or not (`false`).
`emailFollowersOnReply`	boolean	Indicates whether users are notified when other users reply to questions that they're following (`true`) or not (`false`).
`emailOwnerOnPrivateReply`	boolean	Indicates whether users are notified when customer support responds to their questions privately (`true`) or not (`false`).
`emailOwnerOnReply`	boolean	Indicates whether users are notified when other users reply to their questions (`true`) or not (`false`).
`enableAnswerViaEmail`	boolean	Indicates whether users can post answers by replying to email notifications (`true`) or not (`false`). This field is available in API version 29.0 and later.
`enableChatterAnswers`	boolean	Indicates whether the Chatter Answers feature is enabled in the organization (`true`) or not (`false`).
`enableFacebookSSO`	boolean	Indicates whether users can sign in to your Chatter Answers forums with their Facebook logins (`true`) or not (`false`). To enable this feature, you must define and enable a Facebook authentication provider in your organization's security controls and enable Auth Providers in your organization.
`enableInlinePublisher`	boolean	Indicates whether users can filter search results by articles or questions before they post a question to any of your Chatter Answers forums

Field Name	Field Type	Description
		(`true`) or not (`false`). Also, adds `Title` and `Body` fields to questions for easier text input and scanning. This field is available in API version 29.0 and later.
`enableReputation`	boolean	Indicates whether user reputations appear as hover text on their profile pictures (`true`) or not (`false`). Reputation is enabled across all zones. To enable the reputation setting, you must enable Reputation in your organization.
`enableRichTextEditor`	boolean	Indicates whether the rich text editor is enabled for users to format text and upload images when posting questions (`true`) or not (`false`). To enable rich text editor, you must enable Optimize Question Flow.
`facebookAuthProvider`	string	The name of an existing Facebook authentication provider. To implement Facebook Single Sign On for your Chatter Answers forums, you must choose a Facebook authentication provider.
`showInPortals`	boolean	Indicates whether Chatter Answers can be added as a tab to your Customer portal or partner portal (`true`) or not (`false`).

Declarative Metadata Sample Definition

The following is an example `chatteranswers.settings` metadata file:

```
<?xml version="1.0" encoding="UTF-8"?>
<ChatterAnswersSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <emailFollowersOnBestAnswer>true</emailFollowersOnBestAnswer>
    <emailFollowersOnReply>true</emailFollowersOnReply>
    <emailOwnerOnPrivateReply>true</emailOwnerOnPrivateReply>
    <emailOwnerOnReply>true</emailOwnerOnReply>
    <enableChatterAnswers>true</enableChatterAnswers>
    <enableFacebookSSO>true</enableFacebookSSO>
    <enableInlinePublisher>true</enableInlinePublisher>
    <enableReputation>true</enableReputation>
    <enableRichTextEditor>true</enableRichTextEditor>
    <facebookAuthProvider>FacebookAuthProvider</facebookAuthProvider>
    <showInPortals>true</showInPortals>
</ChatterAnswersSettings>
```

The following is an example `package.xml` manifest that references the ChatterAnswersSettings definitions:

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>ChatterAnswers</members>
        <name>Settings</name>
    </types>
    <version>29.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ChatterEmailsMDSettings

Represents an org's settings for Chatter email when Chatter is enabled. This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

The ChatterEmailsMDSettings component appears in the `ChatterEmailsMD.settings` file, and is stored in the `settings` folder. The `.settings` files are different from other named components because there is only one settings file for each settings component.

Version

ChatterEmailsMDSettings components are available in API version 47.0 and later.

Fields

Field Name	Field Type	Description
`enableChatterDigestEmailsApiOnly`	boolean	Indicates whether Chatter digests can be sent via the API, rather than according to the regular schedule, for your org (`true`) or not (`false`).
`enableChatterEmailAttachment`	boolean	Indicates whether attachments can be included on posts to chatter feeds via email replies (`true`) or not (`false`).
`enableCollaborationEmail`	boolean	Indicates whether collaboration email notifications can be sent (`true`) or not (`false`).
`enableDisplayAppDownloadBadges`	boolean	Indicates whether iOS and Android app download badges display in Chatter notifications (`true`) or not (`false`).
`enableEmailReplyToChatter`	boolean	Indicates whether users can reply to chatter notifications through an email response (`true`) or not (`false`).
`enableEmailToChatter`	boolean	Indicates whether users can post to chatter feeds via email (`true`) or not (`false`).
`noQnOwnNotifyOnCaseCmt`	boolean	Indicates whether a user is notified when a question is posted on their case comment (`false`) or not (`true`).
`noQnOwnNotifyOnRep`	boolean	Indicates whether a user is notified when a reply is posted on their question (`false`) or not `true`.

Field Name	Field Type	Description
noQnSubNotifyOnBestR	boolean	Indicates whether a user is notified when a best reply is selected on a question they follow (false) or not (true).
noQnSubNotifyOnRep	boolean	Indicates whether a user is notified when a reply is posted on a question they follow (false) or not (true).

Declarative Metadata Sample Definition

The following is an example of a chatteremailmd.settings metadata file.

```
<?xml version="1.0" encoding="UTF-8"?>
<ChatterEmailsMDSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <enableChatterDigestEmailsApiOnly>false</enableChatterDigestEmailsApiOnly>
    <enableChatterEmailAttachment>false</enableChatterEmailAttachment>
    <enableCollaborationEmail>true</enableCollaborationEmail>
    <enableDisplayAppDownloadBadges>true</enableDisplayAppDownloadBadges>
    <enableEmailReplyToChatter>false</enableEmailReplyToChatter>
    <enableEmailToChatter>true</enableEmailToChatter>
    <noQnOwnNotifyOnCaseCmt>false</noQnOwnNotifyOnCaseCmt>
    <noQnOwnNotifyOnRep>false</noQnOwnNotifyOnRep>
    <noQnSubNotifyOnBestR>false</noQnSubNotifyOnBestR>
    <noQnSubNotifyOnRep>false</noQnSubNotifyOnRep>

</ChatterEmailsMDSettings>
```

Wildcard Support in the Manifest File

The wildcard character * (asterisk) in the package.xml manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ChatterSettings

Represents an org's settings for their Chatter instance when Chatter is enabled for the org. This type extends the Metadata metadata type and inherits its fullName field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

The ChatterSettings component appears in the Chatter.settings file, and is stored in the settings folder. The .settings files are different from other named components because there's only one settings file for each settings component.

Version

ChatterSettings components are available in API version 47.0 and later.

Fields

Field Name	Field Type	Description
`allowChatterGroupArchiving`	boolean	Indicates whether manual and automatic group archiving are allowed on all Chatter groups (`true`) or aren't allowed (`false`). In Setup, `allowChatterGroupArchiving` equates to the Chatter setting **Allow Group Archiving**.
`allowRecordsInChatterGroup`	boolean	Indicates whether records can be associated with groups (`true`), or not (`false`). If groups already have record data, setting this field to `false` doesn't delete it. In Setup, `allowRecordsInChatterGroup` equates to the Chatter setting **Allow Records in Groups**.
`allowSharingInChatterGroup`	boolean	Removed. The setting of this field has no effect on the org. Available in API version 47.0 only.
`enableApprovalRequest`	boolean	Indicates whether Approvals in Chatter are enabled for the org. When the value is `true`, users see approval requests as posts in Chatter feeds. Users can update their own Chatter feeds settings to opt out of receiving approval requests as Chatter posts. When the value is `false`, approval requests aren't posted to Chatter. The default value is `false`. In Setup, `enableApprovalRequest` equates to the Chatter setting **Allow Approvals**.
`enableCaseFeedRelativeTimestamps`	boolean	In Case feeds, indicates whether to use relative (`true`) or absolute (`false`) date and time stamp formats on Case feed items. When the value is `true`, Case feed items show a relative timestamp (for example, `10m ago`). When the value is `true`, users can hover over the relative timestamp to see the absolute. When the value is `false`, Case feed items show an absolute timestamp (for example, `January 7, 2020 at 12:15PM`). When you change this setting, all timestamps in Case feeds reflect that change. The default value is `true`. This field is available in API version 48.0 and later. In Setup, `enableCaseFeedRelativeTimestamps` equates to the Chatter setting **Show relative timestamp**.
`enableChatter`	boolean	Indicates whether Chatter is enabled for your org (`true`) or not (`false`).
`enableChatterEmoticons`	boolean	Indicates whether the automatic conversion of text characters, such as :), into a graphic emoticon is allowed in Chatter (`true`) or isn't allowed (`false`). In Setup, `enableChatter` equates to the Chatter setting **Allow Emoticons**.

Field Name	Field Type	Description
`enableFeedEdit`	boolean	Indicates whether qualified users can edit feed posts and comments (`true`) or not (`false`). Qualified users include: • The author of the post or comment • The person who owns the record that was posted to or commented on • The Chatter or site moderator In Setup, `enableFeedEdit` equates to the Chatter setting **Allow users to edit posts and comments**.
`enableFeedPinning`	boolean	Indicates whether to allow the pinning of posts in a feed (`true`) or not (`false`). When set to `true`: • Authorized users can pin posts to the top of the feed. • The feed supports up to three pinned posts. • Pinned posts stay pinned until they're unpinned. After post pinning is enabled, authorized users include admins and group owners and managers. Admins can also assign post pinning permission through permission sets or user profiles. In Setup, `enableFeedPinning` equates to the Chatter setting **Allow post pinning**.
`enableFeedsDraftPosts`	boolean	Indicates whether draft posts are automatically saved every seven seconds (`true`) or not (`false`). When set to `true`: • Adds the My Drafts feed to the Chatter tab • Saves draft posts automatically every seven seconds • Makes drafts available in the My Drafts feed When the user posts the entry, the draft is automatically removed from the My Drafts feed. In Setup, `enableFeedsDraftPosts` equates to the Chatter setting **Allow draft posts**.
`enableFeedsRichText`	boolean	Indicates whether to use the Rich Text Editor in the Chatter Publisher (`true`) or not (`false`). The rich text editor supports text formats, inline images, hyperlinks, and, when enabled for the org, code snippets. In Setup, `enableFeedsRichText` equates to the Chatter setting **Allow users to compose rich text posts**.
`enableInviteCsnUsers`	boolean	Indicates whether a licensed user can invite customers to private groups that the licensed user owns or manages (`true`) or not (`false`). When the value is set to `true`, licensed users can invite customers who are from outside org email domains. Invited customers can see information only in the groups that they're invited to. They can interact only with members of those groups.

Field Name	Field Type	Description
		In Setup, `enableInviteCsnUsers` equates to the Chatter setting **Allow customer invitations**.
`enableOutOfOfficeEnabledPref`	boolean	Indicates whether to add an Out of Office setting to a user profile page (`true`), or to omit it (`false`). When the value is set to `true`, this option adds a control to user profile pages for setting a personal out-of-office message. In Setup, `enableOutOfOfficeEnabledPref` equates to the Chatter setting **Users can set Out of Office message**.
`enableRichLinkPreviewsInFeed`	boolean	Indicates whether to convert links in posts into embedded videos, images, and article previews (`true`) or not to convert the links (`false`). In Setup, `enableRichLinkPreviewsInFeed` equates to the Chatter setting **Allow Rich Link Previews**.
`enableTodayRecsInFeed`	boolean	Indicates whether to allow the posting of recommendations for using the Salesforce Today app in users' feeds (`true`) or not (`false`). When set to `true`, automatically posts recommendations for using the Salesforce Today app in users' feeds. In Setup, `enableTodayRecsInFeed` equates to the Chatter setting **Allow Today Recommendations**.
`unlistedGroupsEnabled`	boolean	Indicates whether to allow the creation of unlisted groups (`true`) or to prevent their creation (`false`). When the value is set to `true`, users can create unlisted groups. Unlisted groups don't appear on the Groups list page. Membership in unlisted groups is by invitation only. In Setup, `unlistedGroupsEnabled` equates to the Chatter setting **Enable Unlisted Groups**.

Declarative Metadata Sample Definition

The following is an example of a Chatter.settings file.

```
<?xml version="1.0" encoding="UTF-8"?>
 <ChatterSettings xmlns="http://soap.sforce.com/2006/04/metadata">
     <allowChatterGroupArchiving>true</allowChatterGroupArchiving>
     <allowRecordsInChatterGroup>true</allowRecordsInChatterGroup>
     <enableApprovalRequest>true</enableApprovalRequest>
     <enableChatter>true</enableChatter>
     <enableChatterEmoticons>true</enableChatterEmoticons>
     <enableFeedEdit>true</enableFeedEdit>
     <enableFeedsDraftPosts>false</enableFeedsDraftPosts>
     <enableFeedsRichText>true</enableFeedsRichText>
     <enableInviteCsnUsers>true</enableInviteCsnUsers>
     <enableOutOfOfficeEnabledPref>false</enableOutOfOfficeEnabledPref>
     <enableRichLinkPreviewsInFeed>true</enableRichLinkPreviewsInFeed>
     <enableTodayRecsInFeed>true</enableTodayRecsInFeed>
```

```
    <unlistedGroupsEnabled>true</unlistedGroupsEnabled>
</ChatterSettings>
```

The following is an example `package.xml` manifest that references the ChatterSettings definitions:

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>Chatter</members>
        <name>Settings</name>
    </types>
    <version>47.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

CodeBuilderSettings

Represents Code Builder settings. This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

CodeBuilderSettings values are stored in the `CodeBuilder.settings` file in the `settings` folder. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

`CodeBuilderSettings` is available in API versions 58.0 and later.

Fields

Field Name	Field Type	Description
`enableCodeBuilder`	boolean	Indicates whether Code Builder is enabled (true) or not (false). When enabled, you can install and use the generally available (GA) Code Builder package in the org.

Declarative Metadata Sample Definition

The following is an example of the `CodeBuilder.settings` file:

```
<?xml version="1.0" encoding="UTF-8"?>
<CodeBuilderSettings xmlns="http://soap.sforce.com/2006/04/metadata">
```

```
    <enableCodeBuilder>true</enableCodeBuilder>
</CodeBuilderSettings>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

CollectionsDashboardSettings

Represents an org's settings to add the Collections Dashboard application to an org.

Parent Type and Manifest Access

This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all the settings metadata types for the org are accessed using the "Settings" name. See Settings for more details.

File Suffix and Directory Location

`CollectionsDashboardSettings` values are stored in the `CollectionsDashboard.settings` file in the `settings` folder. The `.settings` files are different from other named components, because there's only one settings file for each settings component.

Version

CollectionsDashboardSettings components are available in API version 56.0 and later.

Fields

Field Name	Description
`enableCollectionsDashboard`	**Field Type** boolean **Description** Indicates whether to add the Collections Dashboard application to an org (`true`) or not (`false`). The default value is `false`.

Declarative Metadata Sample Definition

Thisexample shows a sample CollectionsDashboardSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<CollectionsDashboardSettings xmlns="http://soap.sforce.com/2006/04/metadata">
```

```
    <enableCollectionsDashboard>true</enableCollectionsDashboard>
</CollectionsDashboardSettings>
```

Theis example shows a sample `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>CollectionsDashboard</members>
        <name>Settings</name>
    </types>
    <version>64.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

CommunitiesSettings

Represents community settings for an org. Enable digital experiences and workspaces. Manage moderation, guest user and partner settings, and more. This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

`CommunitiesSettings` values are stored in the `Communities.settings` file in the `settings` folder. The `.settings` files are different from other named components, because there's only one settings file for each settings component.

Version

CommunitiesSettings components are available in API version 47.0 and later.

Fields

Field Name	Field Type	Description
`applyLoginPageTypeToEmbeddedLogin`	boolean	When `true`, applies the Experience Cloud site login page type (default, Login Discovery, Experience Builder, or Visualforce) to all Embedded Login implementations. When `false`, applies the username and password login page type to all Embedded Login implementations. For orgs created before the Salesforce Summer '20 release, the default setting is `false`. For new orgs, the default setting is `true`. Available in API version 49.0 and later.
`blockEmbeddedLoginUnknownURLRedirect`	boolean	When `true`, blocks redirects to unknown URLs that are provided in the state parameter of the OAuth response during a server-side callback.

Field Name	Field Type	Description
		Redirects are allowed when the URL is in the same host or domain as the site, or is allow-listed in the Embedded Login salesforce-allowed-domains meta tag. When `false`, all redirects are allowed. For orgs created before the Salesforce Summer '21 release, the default setting is `false`. For new orgs, the default setting is `true`. Available in API version 52.0 and later.
`canModerateAllFeedPosts`	boolean	When `true`, allows moderation features, such as flags and rules, to be set on all feed posts including posts that are visible in Experience Cloud sites. When set to `false`, only feed posts in sites can be moderated. Default is `false`.
`canModerateInternalFeedPosts`	boolean	When `true`, allows moderation features, such as flags and rules, to be set on record feed posts created by internal users. Such posts can also be visible in multiple sites. Default is `false`.
`embeddedVisualforcePages`	boolean	When `true`, allows links to Visualforce pages from other Visualforce pages in Salesforce via the API. Default is `false`. Available in API version 48.0 and later.
`enableCommunityWorkspaces`	boolean	When `true`, allows admins to enable Experience Workspaces. Available in API version 48.0 and later.
`enableCspContactVisibilityPref`	boolean	When `true`, allows users to see contacts from private accounts that they have read access to, when the contact is controlled by the parent record. Available in API version 48.0 and later.
`enableCspNotesOnAccConPref`	boolean	When `true`, allows customer users to access notes and attachments associated with accounts and contacts. Available in API version 48.0 and later.
`enableEnablePRM`	boolean	When `true`, allows admins to enable partner users. Available in API version 48.0 and later.
`enableExternalAccHierPref`	boolean	When `true`, enables the External Account Hierarchy object. Available in API version 48.0 and later.
`enableGuestRecordReassignOrgPref`	boolean	Deprecated in API version 63.0 and later. When `true`, allows admins to set a default owner for records created by guest users. Available in API version 48.0 through 63.0.
`enableGuvSecurityOptOutPref`	boolean	When `true`, guest user visibility can be turned off. Available in API version 49.0 and later.
`enableInviteChatterGuestEnabled`	boolean	When `true`, allows guest users to be invited to use Chatter. Available in API version 48.0 and later.
`enableNetPortalUserReportOpts`	boolean	When `true`, allows external users in Experience Cloud sites, with permission, to run reports. Available in API version 48.0 and later.

Field Name	Field Type	Description
enableNetworksEnabled	boolean	When true, allows users to enable digital experiences. Available in API version 47.0 and later.
enableOotbProfExtUserOpsEnable	boolean	When true, allows use of standard external profiles for self-registration and user creation. Available in API version 48.0 and later.
enablePreventBadgeGuestAccess	boolean	When true, hides badges from guest users in Experience Builder sites. Available in API version 53.0 and later.
enablePowerCustomerCaseStatus	boolean	When true, allows users with Customer Community Plus licenses to change case status. Available in API version 48.0 and later.
enablePRMAccRelPref	boolean	When true, enables Account Relationship object and Account Relationship Data Sharing Rule setup options. Available in API version 48.0 and later.
enableRelaxPartnerAccountFieldPref	boolean	When true, allows editing for partner account fields on and opportunities and leads. Available in API version 48.0 and later.
enableUnsupportedBrowserModalPref	boolean	When true, warnings about unsupported browsers are displayed in Experience Cloud sites. Available in API version 48.0 and later.
enableUsernameUniqForOrgPref	boolean	When true, username uniqueness is set at the org level. Available in API version 48.0 and later.

Declarative Metadata Sample Definition

The following is an example of a CommunitiesSettings component.

```
<CommunitiesSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <canModerateAllFeedPosts>true</canModerateAllFeedPosts>
    <canModerateInternalFeedPosts>true</canModerateInternalFeedPosts>
</CommunitiesSettings>
```

The following is an example package.xml that references the previous definition.

```
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
<types>
<members>Communities</members>
<name>Settings</name>
</types>
<version>47.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character * (asterisk) in the package.xml manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

CompanySettings

Represents global settings that affect multiple features in your organization. This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

Declarative Metadata File Suffix and Directory Location

CompanySettings values are stored in a single file named `Company.settings` in the `settings` directory of the corresponding package directory. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

Company Profile Settings are available in API version 27.0 and later.

Fields

Field Name	Field Type	Description
`enableCustomFiscalYear`	boolean	If a custom fiscal period is set up, this field is used to determine whether the custom fiscal period is used for forecasts. If `true`, the custom fiscal period is used. If `false` (default), standard periods are used. Available in API version 47.0 and later.
`fiscalYear`	FiscalYearSetting	The organization's fiscal year setting based on year and start month. Not available if Custom Fiscal Year or Forecasts (Classic) is enabled. When changing fiscal year settings, quotas and adjustments can be purged. For example changing your start month results in purging this data.

FiscalYearSetting

Represents your organization's fiscal year setting.

Field	Field Type	Description
`fiscalYearNameBasedOn`	string	This field is used to determine the fiscal year name. Valid values are `endingMonth` or `startingMonth`. For example, if your fiscal year starts in April 2012 and ends in March 2013, and this value is: • `endingMonth`, then 2013 is used for the fiscal year name. • `startingMonth`, then 2012 is used for the fiscal year name.
`startMonth`	string	The month on which the fiscal year is based.

Declarative Metadata Sample Definition — Fiscal Year Setting

A sample XML definition of a fiscal year setting is shown below. Note that this example is supported in API version 27.0 and later.

```
<?xml version="1.0" encoding="UTF-8"?>
<CompanySettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <fiscalYear>
        <fiscalYearNameBasedOn>endingMonth</fiscalYearNameBasedOn>
        <startMonth>January</startMonth>
    </fiscalYear>
</CompanySettings>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ConnectedAppSettings

Represents settings for connected apps. This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

ConnectedAppSettings values are stored in a single file named `ConnectedApp.settings` in the `settings` directory. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

ConnectedAppSettings components are available in API version 47.0 and later.

Fields

Field Name	Field Type	Description
`enableAdminApprovedAppsOnly`	boolean	If `false` (default), any connected app can call the Salesforce API. If `true`, only apps that have been approved or installed by the admin can call the Salesforce API. To access this field, you must contact Salesforce Customer Support to enable API Access Control.
`enableAdminApprovedAppsOnlyForExternalUser`	boolean	If `false` (default), authenticated customers or partners can use any unblocked connected app to access the Salesforce API. If `true`, authenticated customers and partners can't access the Salesforce API unless they use a connected app that is installed in the org and unblocked. Install and unblock connected apps on the Connected Apps OAuth Usage page. To access this field, you must contact Salesforce Customer Support to enable API Access Control.
`enableSkipUserProvisioningWizardWelcomePage`	boolean	If `false` (default), the User Provisioning Wizard Welcome page shows up when you access the wizard. To skip the welcome page in the future,

Field Name	Field Type	Description
		you can select **Do not show me this next time**. If `true`, the Welcome page doesn't show up the next time that you access the wizard.

Declarative Metadata Sample Definition

The following is an example of a ConnectedAppSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ConnectedAppSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <enableAdminApprovedAppsOnly>false</enableAdminApprovedAppsOnly>

<enableAdminApprovedAppsOnlyForExternalUser>false</enableAdminApprovedAppsOnlyForExternalUser>


<enableSkipUserProvisioningWizardWelcomePage>true</enableSkipUserProvisioningWizardWelcomePage>
</ConnectedAppSettings>
```

The following is an example `package.xml` manifest that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>ConnectedApp</members>
        <name>Settings</name>
    </types>
    <version>47.0</version>
</Package>
```

ContentSettings

Represents content settings for an org. This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

`ContentSettings` values are stored in the `contentsettings.settings` file in the `settings` folder. The `.settings` files are different from other named components, because there's only one settings file for each settings component.

Version

ContentSettings components are available in API version 48.0 and later.

Fields

Field Name	Field Type	Description
`enableChatterFileLink`	boolean	When `true`, allows users to share files via links. When set to `false`, users can't use file sharing. Default is `true`.

Field Name	Field Type	Description
enableCMSC2CConnections	boolean	When true, allows org to enable Salesforce CMS Cloud to Cloud Connections.
enableContent	boolean	When true, allows org to enable Content.
enableContentAutoAssign	boolean	When true, allows org to auto assign Content feature licenses to users.
enableContentDistForPortalUsers	boolean	When true, allows portal users to create Content Deliveries for managed files in a library.
enableContentDistPwOptionsBit1	boolean	When true, allow using with ContentDistPasswordOptionsBit2 to set up one of three possible delivery security options.
enableContentDistPwOptionsBit2	boolean	When true, allow using with ContentDistPasswordOptionsBit1 to set up one of three possible delivery security options.
enableContentDistribution	boolean	When true, allows the Content Delivery user permission to be enabled for users. Content deliveries let users create links to share files externally, with optional security settings.
enableContentSupportMultiLanguage	boolean	When true, enables content to support multiple languages.
enableContentWorkspaceAccess	boolean	When true, content libraries are visible in the API and UI for users who have read access to libraries, even if they don't have access to the original Salesforce CRM Content app.
enableDeleteFileInContentPacks	boolean	When true, enables an org preference that allows a file's owner to delete the file, which is included in one or more content packs. The default setting is based on the org.
enableFileShareSetByRecord	boolean	When true, files shared to records default to Set by Record.
enableFilesUsrShareNetRestricted	boolean	When true, files respect user sharing settings. Files shared with users with **SharedUsers** visibility are only accessible to users who are members of the Experience Cloud site the file was created in.
enableJPGPreviews	boolean	When true, attempts to use other SVG alternative formats such as JPG as preview images.
enableLibraryManagedFiles	boolean	When true, controls the ability to publish files created in Chatter with a Content Library (ContentWorkspace). The library can manage the file.
enableShowChatterFilesInContent	boolean	When true, allows users to search for Chatter files in content.
enableSiteGuestUserToUploadFiles	boolean	When true, site guest users can upload files.
enableUploadFilesOnAttachments	boolean	When true, shows the **New File** button on the Attachments related lists to upload files, rather than legacy Attachments.
setValidContentTypeForAtchDocDownload	boolean	When true, enables an org preference that controls whether the Content-Type HTTP response header is set to a valid content type during file and attachment downloads. If false the response header is set

Field Name	Field Type	Description
		to the value that the user provided during file upload. Values provided by the user can be invalid. If the content type provided by the user isn't a valid content type, the system tries to determine a valid content type based on the file name extension. This field is available in API version 50.0 and later.
`skipContentAssetTriggers`	boolean	When `true`, disables content trigger execution and custom validation on content assets.
`skipContentAssetTriggersOnDeploy`	boolean	When `true`, disables content trigger execution when deploying content assets.

Declarative Metadata Sample Definition

The following is an example of a Content.settings file.

```
<ContentSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <enableChatterFileLink>true</enableChatterFileLink>
    <enableCMSC2CConnections>true</enableCMSC2CConnections>
    <enableContent>true</enableContent>
    <enableContentAutoAssign>true</enableContentAutoAssign>
    <enableContentDistForPortalUsers>true</enableContentDistForPortalUsers>
    <enableContentDistribution>true</enableContentDistribution>
    <enableContentSupportMultiLanguage>false</enableContentSupportMultiLanguage>
    <enableContentWorkspaceAccess>true</enableContentWorkspaceAccess>
    <enableFileShareSetByRecord>true</enableFileShareSetByRecord>
    <enableFilesUsrShareNetRestricted>true</enableFilesUsrShareNetRestricted>
    <enableJPGPreviews>true</enableJPGPreviews>
    <enableLibraryManagedFiles>true</enableLibraryManagedFiles>
    <enableShowChatterFilesInContent>true</enableShowChatterFilesInContent>
    <enableSiteGuestUserToUploadFiles>true</enableSiteGuestUserToUploadFiles>
    <enableUploadFilesOnAttachments>true</enableUploadFilesOnAttachments>
    <skipContentAssetTriggers>true</skipContentAssetTriggers>
    <skipContentAssetTriggersOnDeploy>true</skipContentAssetTriggersOnDeploy>
</ContentSettings>
```

The following is an example `ContentSettings.xml` manifest that references the ContentSettings definitions:

```
<?xml version="1.0" encoding="UTF-8"?>
<ContentSettings xmlns=
    <types>
        <members>Content</members>
        <name>Settings</name>
    </types>
    <version>48.0</version>
</ContentSettings>
```

Wildcard Support in the Manifest File

The wildcard character * (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ContractSettings

Represents contract settings.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

There is one contract settings file stored in a file named `Contract.settings` in the `settings` directory. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

ContractSettings is available in API version 27.0 and later.

Fields

Field Name	Field Type	Description
`autoCalculateEndDate`	boolean	Indicates whether the end date of a contract is automatically calculated (`true`) or not (`false`).
`notifyOwnersOnContractExpiration`	boolean	Indicates whether account and contract owners are automatically sent email notifications when a contract expires (`true`) or not (`false`).

Declarative Metadata Sample Definition

This is a sample contract settings file.

```
<?xml version="1.0" encoding="UTF-8"?>
<ContractSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <autoCalculateEndDate>true</autoCalculateEndDate>
    <notifyOwnersOnContractExpiration>false</notifyOwnersOnContractExpiration>
</ContractSettings>
```

Wildcard Support in the Manifest File

The wildcard character * (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ConversationalIntelligenceSettings

Represents the org's Einstein Conversation Insights settings, such as whether Einstein Conversation Insights is enabled. Einstein Conversation Insights lets you analyze your rep's call recordings, and gives you the insights you need to optimize every call.

Parent Type and Manifest Access

This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all the settings metadata types for the org are accessed using the "Settings" name. See Settings for more details.

File Suffix and Directory Location

ConversationalIntelligenceSettings values are stored in the `ConversationalIntelligence.settings` file in the `settings` folder.

Version

ConversationalIntelligenceSettings components are available in API version 49.0 and later.

Fields

Field Name	Description
`enableCallCoaching`	**Field Type** boolean **Description** Indicates whether Einstein Conversation Insights is enabled (`true`) or not (`false`). The default value is `false`. Available in API version 49.0 and later.
`enableCallCoachingZoom`	**Field Type** boolean **Description** Indicates whether Zoom video calls are enabled for Einstein Conversation Insights (`true`) or not (`false`). The default value is `false`. Available in API version 51.0 and later.
enableCallExplorer	**Field Type** boolean **Description** Indicates whether Call Explorer is enabled (`true`) or not (`false`). The default value is `false`. Available in API version 60.0 and later.
enableCallSummarization	**Field Type** boolean

Field Name	Description
	Description Indicates whether Call Summary is enabled (`true`) or not (`false`). The default value is `false`. Available in API version 59.0 and later.
enableConversationMining	**Field Type** boolean **Description** Indicates whether Einstein Conversation Mining is enabled (`true`) or not (`false`). The default value is `false`. Available in API version 61.0 and later.
`enableDiarizationPref`	**Field Type** boolean **Description** Indicates whether optimal speaker separation is enabled (`true`) or not (`false`). The default value is `false`. Available in API version 58.0 and later. Optimal speaker separation lets Einstein Conversation Insights use acoustic features of speaker voices to separate an audio stream into separate segments.
enableGenerativeConvInsights	**Field Type** boolean **Description** Indicates whether Generative Conversation Insights is enabled (`true`) or not (`false`). The default value is `false`. Available in API version 61.0 and later.
enableManualUpload	**Field Type** boolean **Description** Indicates whether Video Call Upload is enabled (`true`) or not (`false`). The default value is `false`. Available in API version 62.0 and later. Reserved for future use.
`enableOpptyMatching`	**Field Type** boolean **Description** Indicates whether voice and video calls are related to opportunities automatically (`true`) or not (`false`). The default value is `false`. Available in API version 53.0 and later.
enableRealtimeInsights	**Field Type** boolean

Field Name	Description
	Description Indicates whether Realtime Insights is enabled (`true`) or not (`false`). The default value is `false`. Available in API version 62.0 and later.
`enableUnifiedActivities`	**Field Type** boolean **Description** Indicates whether Activity 360 Reporting is enabled (`true`) or not (`false`). The default value is `false`. Available in API version 58.0 and later.

Declarative Metadata Sample Definition

The following is an example of a ConversationalIntelligenceSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ConversationalIntelligenceSettings xmlns="http://soap.sforce.com/2006/04/metadata">
 <enableCallCoaching>true</enableCallCoaching>
</ConversationalIntelligenceSettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
 <types>
  <members>ConversationalIntelligence</members>
  <name>Settings</name>
 </types>
 <version>49.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ConversationChannelDefinition

Represents the conversation channel definition that's implemented for Interaction Service for Bring Your Own Channel for Messaging and Bring Your Own Channel for CCaaS messaging channels. This object is available in API version 60.0 and later.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

ConversationChannelDefinition components have the suffix `.ConversationChannelDefinition` and are stored in the `conversationChannelDefinitions` folder.

Version

ConversationChannelDefinition components are available in API version 60.0 and later.

Special Access Rules

Interaction service must be configured. Access to tooling objects requires Salesforce administrator privileges or the Customize Application permission.

Fields

Field Name	Description
`connectedAppOauthLink`	**Field Type** string **Description** DO NOT SET OR CHANGE THIS VALUE. This value is automatically generated. This field represents the OAuth link for the connected app if the connectedAppType value is `Partner`. This is a string identifier to the connected app containing the partner Org ID and the consumer ID minus the key prefixes. Used to identify the connected app to use for the channel definition and channel at runtime.
`connectedAppType`	**Field Type** CustomChannelConnectedAppType **Description** The owner of the connected app used to manage authentication between Salesforce Interaction Service and the Messaging or CCaaS partner's system. Possible values are: • `Partner` • `Customer` The default value is `Partner`. If set to `Partner`, the partner creates the connected app and includes it in their managed package. If set to `Customer`, the admin creates the connected app. Available in API version 62.0 and later.
`consentOwner`	**Field Type** ConsentOwner (enumeration of type string) **Description** The system the customer uses to manage consent levels.

Field Name	Description
	Possible values are: • `Partner` • `Salesforce` The default value is `Salesforce`. For example, if set to *`Salesforce`*, consent levels are managed by the Salesforce system. If set to *`Partner`*, consent levels are managed by the partner's telephony system. For Bring Your Own Channel for Messaging, this value must be set to *`Salesforce`*.
`conversationVendorInfo`	**Field Type** string **Description** The *`ConversationVendorInfo.developerName`* used to link this record to the ConversationVendorInfo record. For example, PartnerName.
`customEventChnlAddrIdField`	**Field Type** string **Description** The mapping field that points to the custom field used to point to the `ChannelAddressIdentifier` field. This field is available in API version 60.0 and earlier. Use a combination of `customEventTypeField` and `customEventPayloadField` in API version 61.0 and later instead.
`customEventPayloadField`	**Field Type** string **Description** Required. The mapping field that points to the custom field used to point to the `Payload` field in the format *`<orgNamespace>__<CustomFieldName>__`*c. This is the API name of the custom Payload field in the custom platform event. For example, devorg__Payload__c.
`customEventRecipientField`	**Field Type** string **Description** The mapping field that points to the custom field used to point to the Recipient field. This field is available in API version 60.0 and earlier. Use a combination of `customEventTypeField` and `customEventPayloadField` in API version 61.0 and later instead.
`customEventTypeField`	**Field Type** string

Field Name	Description
	Description The mapping field that points to the custom field used to point to the Platform event type (EventType) field, in the format *<orgNamespace>__<CustomFieldName>__*c. This is the API name of the custom EventType field in the custom platform event. For example, devorg__EventType__c.
`customIcon`	**Field Type** string **Description** For Bring Your Own Channel for Messaging and Bring Your Own Channel for CCaaS, this field represents the name of the status resource image used to identify the channel integration, such as a channel logo. For the best results, set the image size to 50px x 50px and save the image in SVG file format. This field is optional. This field is a relationship field. Available in API version 61.0 and later.
`customPlatformEvent`	**Field Type** string **Description** Required. The API name of the custom platform event created for the Interaction Service API in the format *<orgNamespace>__<CustomPlatformEventName>__*e. For example, devorg__TestEvent__e.
`customerConnectedAppOauthLink`	**Field Type** string **Description** DO NOT SET OR CHANGE THIS VALUE. This value is automatically generated. This field represents the OAuth link for the connected app created by an admin if the ConnectedAppType is `Customer`. Available in API version 62.0 and later.
`developerName`	**Field Type** string **Description** The unique name of the custom metadata type object in the API in the format *<Prefix>_<ConversationChannelDefinition>*, where *Prefix* matches the prefix you gave to the name of the interaction service connected app. For example, Partner1_ChannelDefinition1, where Partner1 is the prefix and ChannelDefinition1 is the given name.
`isConferenceSupported`	**Field Type** boolean

Field Name	Description
	Description Indicates whether the partner supports conferencing for Bring Your Own Channel (`true`), or not (`false`). With conferencing, more than two participants are allowed in a messaging session. The default is `false`. This field is available in API version 64.0 and later.
`IsRoutingWorkResultEnabled`	**Field Type** boolean **Description** Indicates whether the Routing Work Result event is sent as a Custom Platform event or not. The default value is `false`. A partner can change the field value to `true` in API version 65.0 and later.
`isInboundReceiptsEnabled`	**Field Type** boolean **Description** Indicates whether the partner supports read receipts and delivery receipts for inbound messages (`true`) or whether the partner doesn't support these inbound acknowledgements and the functionality is hidden from the Salesforce admin in the Messaging settings (`false`). The default value is `false`. This field is available in API version 63.0 and later.
`isTypingIndicatorDisabled`	**Field Type** boolean **Description** Indicates whether the partner doesn't support typing indicators for outbound messages and the functionality is hidden from the Salesforce admin in the Messaging settings (`true`) or whether outbound typing indicators are supported by the partner (`false`). The default value is `false`, meaning the outbound typing indicator feature is supported by default. To disable the outbound typing indicator feature, set this value to `true`. This field is available in API version 63.0 and later.
`masterLabel`	**Field Type** string **Description** The namespace prefix that is associated with this object. Each Developer Edition org that creates a managed package has a unique namespace prefix. Limit: 15 characters. You can refer to a component in a managed package by using the *`namespacePrefix__componentName`* notation. The namespace prefix can have one of the following values.

Field Name	Description
	• In Developer Edition orgs, `NamespacePrefix` is set to the namespace prefix of the org for all objects that support it, unless an object is in an installed managed package. In that case, the object has the namespace prefix of the installed managed package. This field's value is the namespace prefix of the Developer Edition org of the package developer. • In orgs that are not Developer Edition orgs, `NamespacePrefix` is set only for objects that are part of an installed managed package. All other objects have no namespace prefix. `NamespacePrefix` is null if the publisher is Salesforce.
`maxParticipantsForCnfrOverride`	**Field Type** int **Description** Specifies the limit for how many participants can be in a messaging conference. If set, this field overrides the platform limit for the number of participants in a conference. If not set, the limit defaults to the messaging platform limit of how many participants can be in a messaging conference at one time. This field is available in API version 64.0 and later.
`routingOwner`	**Field Type** RoutingOwner (enumeration of type string) **Description** The system the customer uses to manage routing for Bring Your Own Channel for Messaging or Bring Your Own Channel for CCaaS. Possible values are: • `Partner` • `Salesforce` The default value is `Salesforce`. For example, if set to *`Salesforce`*, routing is managed by the Salesforce system. If set to *`Partner`*, routing is managed by the partner's telephony or Contact Center as a Service (CCaaS) system. For Bring Your Own Channel for Messaging, this value must be set to *`Salesforce`*.
`supportsCustomChannelParameters`	**Field Type** boolean **Description** Indicates whether admins can configure custom parameters and parameter mappings for messaging channels. Custom parameters and parameter mappings are used to pass additional information at runtime to Omni-Channel flows. The default value is false. This field is available in API version 61.0 and later.

Field Name	Description
`supportsDoubleOptInConsent`	**Field Type** boolean **Description** Indicates whether the channel supports (*`true`*) the Double Opt-In consent level. The default value is *`false`*. If set to true, then `capabilitiesSupportsExplicitConsent` must also be set to true. This field is optional and isn't supported for Bring Your Own Channel for Messaging. It's only supported for Bring Your Own Channel for CCaaS.
`supportsExplicitConsent`	**Field Type** boolean **Description** Indicates whether the channel supports (*`true`*) the Explicit Opt-In consent level. This field is optional.
`supportsImplicitConsent`	**Field Type** boolean **Description** Indicates whether the channel supports (*`true`*) the Implicit Opt-In consent level. This value is required and must always be set to true. The default value is false.
`supportsIsoCountryCode`	**Field Type** boolean **Description** Indicates whether the channel supports (*`true`*) ISO country codes. The default value is false.
`supportsKeywords`	**Field Type** boolean **Description** Indicates whether the channel supports (*`true`*) keywords. The default value is false.

Declarative Metadata Sample Definition

The following is an example of a ConversationChannelDefinition component.

The following is an example `package.xml` that references the previous definition.

CurrencySettings

Represents an organization's currency settings, including supporting multiple currencies and currency effective dates. This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

CurrencySettings values are stored in the `Currency.settings` file in the `settings` directory. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

CurrencySettings is available in API version 47.0 and later.

Fields

Field Name	Field Type	Description
`enableCurrencyEffectiveDates`	boolean	Indicates whether effective dated currency is enabled (`true`) or not (`false`). This field has a default value of `false`. To enable this preference, `enableMultiCurrency` must be set to `true`
`enableCurrencySymbolWithMultiCurrency`	boolean	Indicates whether the currency symbol (`true`) or ISO code (`false`) displays in multi-currency orgs. This field has a default value of `false`. This field has no effect if `enableMultiCurrency` is set to `false`.
`enableMultiCurrency`	boolean	Indicates whether multiple currencies are enabled (`true`) or not (`false`). This field has a default value of `false`. Note: After set to `true`, this field can't be set to `false`. See Considerations for Enabling Multiple Currencies for more information.
`isMultiCurrencyActivationAllowed`	boolean	Deprecated in API version 49.0 and later. Regardless of the value in this field, a Salesforce admin can activate multiple currencies. In API version 48.0 and earlier, if Customizable Forecasting was enabled, this field indicated whether Salesforce Customer Support could activate multiple currencies (`true`) or the feature couldn't be activated (`false`). This field is only visible if multiple currencies are disabled. It has a default of `false` to provide an extra layer of protection against accidentally enabling multiple currencies when Customizable Forecasting was enabled. In API version 48.0 and earlier, customers with Customizable Forecasting enabled in their orgs had to contact Salesforce Customer Support to activate multiple currencies. Customers set this field to `true` when Salesforce Customer Support requested that they do so to validate their request to activate multiple currencies. Note: Customizable Forecasting was retired in Summer '20. Users can't access the Customizable Forecasting feature and its underlying data via the user interface or API. To predict sales revenue and quantities from your opportunity pipeline, use Salesforce Forecasting.

Field Name	Field Type	Description
isParenCurrencyConvDisabled	boolean	Indicates whether parenthetical currency conversion is disabled (true) or enabled (false). This field has a default value of true. When this field is set to false, Salesforce displays converted currency amounts to users whose personal currency differs from the currency of the record they're viewing.

Declarative Metadata Sample Definition

The following is an example of a CurrencySettings file.

```
<?xml version="1.0" encoding="UTF-8"?>
<CurrencySettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <enableCurrencyEffectiveDates>false</enableCurrencyEffectiveDates>
    <enableCurrencySymbolWithMultiCurrency>false</enableCurrencySymbolWithMultiCurrency>
    <enableMultiCurrency>false</enableMultiCurrency>
    <isMultiCurrencyActivationAllowed>false</isMultiCurrencyActivationAllowed>
    <isParenCurrencyConvDisabled>false</isParenCurrencyConvDisabled>
</CurrencySettings>
```

The following is an example package.xml that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>Currency</members>
        <name>Settings</name>
    </types>
    <version>47.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character * (asterisk) in the package.xml manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

CustomAddressFieldSettings

Represents the settings for custom address fields.

Parent Type and Manifest Access

This type extends the Metadata metadata type and inherits its fullName field.

In the package manifest, all the settings metadata types for the org are accessed using the "Settings" name. See Settings for more details.

File Suffix and Directory Location

`CustomAddressFieldSettings` values are stored in the `CustomAddressField.settings` file in the `settings` folder. The `.settings` files are different from other named components, because there is only one settings file for each settings component.

Version

CustomAddressFieldSettings components are available in API version 55.0 and later.

Fields

Field Name	Field Type	Description
`enableCustomAddressField`	boolean	Indicates whether the `Address` Field Type is available for custom fields (`true`) or not (`false`). The default value is `false`. Custom Address Fields can't be disabled. When `enableCustomAddressField` is set to `true`, you can't change the value to `false`. Note: Before you set this field to `true`, review Custom Address Fields Requirements and Limitations in Salesforce Help.

Declarative Metadata Sample Definition

The following is an example of a CustomAddressFieldSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<CustomAddressFieldSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <enableCustomAddressField>true</enableCustomAddressField>
</CustomAddressFieldSettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>CustomAddressField</members>
        <name>Settings</name>
    </types>
    <version>55.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

DataDotComSettings

Represents the org's Data.com settings. This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

DataDotComSettings values are stored in the `DataDotCom.settings` file in the `settings` folder.

Version

DataDotComSettings components are available in API version 47.0 and later.

Fields

Field Name	Field Type	Description
`enableAccountExportButtonOff`	boolean	Indicates whether Account Export to Excel is enabled for Prospector (`true`) or not (`false`). Default value is `false`.
`enableAccountImportButtonOff`	boolean	Indicates whether Account Import to CRM is enabled for Prospector (`true`) or not (`false`). Default value is `false`.
`enableAllowDupeContactFromLead`	boolean	Indicates whether Prospector Lead Import Duplicate Check is enabled (`true`) or not (`false`). Default value is `false`.
`enableAllowDupeLeadFromContact`	boolean	Indicates whether Prospector Contact Import Duplicate Check is enabled (`true`) or not (`false`). Default value is `false`.
`enableCleanUpgradeRequested`	boolean	This field is no longer in use.
`enableContactExportButtonOff`	boolean	Indicates whether Contact Export to Excel is enabled for Prospector (`true`) or not (`false`). Default value is `false`.
`enableContactImportButtonOff`	boolean	Indicates whether Contact Import to CRM is enabled for Prospector (`true`) or not (`false`). Default value is `false`.

Declarative Metadata Sample Definition

The following is an example of a DataDotComSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<DataDotComSettings xmlns="http://soap.sforce.com/2006/04/metadata">
 <enableAccountExportButtonOff>true</enableAccountExportButtonOff>
 <enableAccountImportButtonOff>true</enableAccountImportButtonOff>
 <enableAllowDupeContactFromLead>true</enableAllowDupeContactFromLead>
 <enableAllowDupeLeadFromContact>true</enableAllowDupeLeadFromContact>
 <enableContactExportButtonOff>true</enableContactExportButtonOff>
 <enableContactImportButtonOff>true</enableContactImportButtonOff>
</DataDotComSettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
 <types>
  <members>DataDotCom</members>
  <name>Settings</name>
 </types>
 <version>29.0</version>
</Package>
```

DataImportManagementSettings

Represents an org's contact and leads import settings.

Parent Type and Manifest Access

This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all the settings metadata types for the org are accessed using the "Settings" name. See Settings for more details.

File Suffix and Directory Location

`DataImportManagementSettings` values are stored in the `DataImportManagement.settings` file in the `settings` folder. The `.settings` files are different from other named components, because there is only one settings file for each settings component.

Version

DataImportManagementSettings components are available in API version 57.0 and later.

Special Access Rules

DataImportManagementSettings is available when your org enables the DataImportManagement permission, which is only available for particular editions.

Fields

Field Name	Description
`enableEasyImport`	**Field Type** boolean **Description** Indicates whether Basic Data Import is enabled (`true`) or not (`false`). When `true`, users are guided step by step to select how they want to import contacts and leads to Salesforce.

Declarative Metadata Sample Definition

The following is an example of a DataImportManagementSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<DataImportManagementSettings xmlns="http://soap.sforce.com/2006/04/metadata">
          <enableEasyImport>true</enableEasyImport>
</DataImportManagementSettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
          <types>
               <members>DataImportManagement</members>
               <name>Settings</name>
          </types>
          <version>57.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

DeploymentSettings

Represents the settings affecting how deployments behave in the org. This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

DeploymentSettings values are stored in the `Deployment.settings` file in the `settings` directory. The `.settings` files are different from other named components, because there is only one settings file for each settings component.

Version

DeploymentSettings components are available in API version 47.0 and later.

Fields

Field Name	Field Type	Description
`doesSkipAsyncApexValidation`	boolean	Indicates whether deployments from this org skip asynchronous Apex validations (`true`) or not (`false`). The default value is `true`. Set this field to `false` when an Apex class in the package you're deploying is used by an Apex batch job that could run during the

Field Name	Field Type	Description
		deployment. The deployment of a package containing an Apex class that is used by a running batch job fails without validation.

Declarative Metadata Sample Definition

The following is an example of a DeploymentSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<DeploymentSettings xmlns="http://soap.sforce.com/2006/04/metadata">
   <doesSkipAsyncApexValidation>true</doesSkipAsyncApexValidation>
</DeploymentSettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
   <types>
      <members>Deployment</members>
      <name>Settings</name>
   </types>
   <version>47.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

DevHubSettings

Represents Dev Hub settings.

Parent Type and Manifest Access

This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all org settings metadata types are accessed using the Settings name. See Settings for more details.

File Suffix and Directory Location

DevHubSettings values are stored in the `DevHub.settings` file in the `settings` folder. The `.settings` files are different from other named components because there is only one settings file for each settings component.

Version

DevHubSettings are available in API version 47.0 and later.

Fields

Field Name	Field Type	Description
`enableDevOpsCenterGA`	boolean	Indicates whether DevOps Center is enabled: `true` or `false`. When enabled, you can install and use the generally available (GA) DevOps Center package in the org. Available in API version 56.0 and later.
`enableDevOpsCenter`	boolean	(Deprecated) Indicates whether DevOps Center beta is enabled: `true` or `false`. When enabled, you can install and use the DevOps Center beta package in the org.
`enablePackaging2`	boolean	Indicates whether unlocked and second-generation managed packaging is enabled: `true` or `false`. To enable `enablePackaging2`, first enable `enableScratchOrgManagementPref`.
`enableScratchOrgManagementPref`	boolean	Indicates whether Dev Hub is enabled: `true` or `false`. When enabled, a Dev Hub allows you to create and manage scratch orgs.
`enableScratchOrgSnapshotPref`	boolean	Indicates whether Scratch Org Snapshots is enabled: `true` or `false`. When enabled, you can create snapshots of a fully configured scratch org. A snapshot is a point-in-time copy of a scratch org that you can use to create additional scratch orgs. Available in API version 61.0 and later.
`enableShapeExportPref`	boolean	Indicates whether Org Shape is enabled: `true` or `false`. When enabled, you can create org shapes as the basis for scratch orgs. Available in API version 55.0 and later.

Declarative Metadata Sample Definition

The following is an example of a DevHubSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<DevHubSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <enablePackaging2>true</enablePackaging2>
    <enableScratchOrgManagementPref>true</enableScratchOrgManagementPref>
</DevHubSettings>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

DocumentGenerationSetting

Represents an org's settings for automatic document generation from templates. This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

DocumentGenerationSetting components have the suffix `documentGenerationSetting` and are stored in the `documentGenerationSettings` folder. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

DocumentGenerationSetting components are available in API version 52.0 and later.

Special Access Rules

DocumentGenerationSetting is available if your org has the DocGen platform license and related addon and user licenses.

Server-side document generation isn't enabled by default, and selecting `isServerSideDocGenEnabled` isn't sufficient. To enable this feature, see the Request Access to Server-Side Document Generation knowledge article.

Fields

Field Name	Field Type	Description
`documentTemplateLibraryName`	String	Required. The name of the library that stores the document templates to which this setting applies.
`generationMechanism`	GenerationMechanism (enumeration of type string)	Specifies how and where a document is generated. Valid values are: • `ClientSide`—Generates documents in the browser with an optional preview. • `ServerSide`—Generates documents on the server and attaches them to the objects for which they're generated. The default is `ClientSide`.
`guestAccessNamedCredential`	String	Specifies the named credential that lets guest users generate documents.
`inProgDocGenRqstTmot`	Integer	The time interval (in hours) after which InProgress`DocumentGenerationProcess` records (single point or batch request) are terminated and the request status is updated to Failed. Specify a valid whole number between 1 and 24. The default value is set to 6.
`isInProgRqstTmotEnab`	Boolean	Indicates whether the InProgress DocGen Request Time Out (hrs) is enabled (`true`) or not (`false`).
`isServerSideDocGenEnabled`	Boolean	Enables server-side document generation if the prerequisite license is present in the org.

Field Name	Field Type	Description
masterLabel	String	Required. Specifies a name for the setting, such as DocGen.
previewType	PreviewType (enumeration of type string)	Specifies the format of previews of generated documents. Valid values are: • PDF—Displays how the generated document looks in PDF format. • Thumbnail—Displays a miniature representation of the generated document. The default is PDF.

Declarative Metadata Sample Definition

The following is an example of a DocumentGenerationSetting component.

```
<?xml version="1.0" encoding="UTF-8"?>
<DocumentGenerationSetting xmlns="http://soap.sforce.com/2021/10/metadata">
   <documentTemplateLibraryName>DocgenDocumentTemplateLibrary</documentTemplateLibraryName>

    <masterLabel>DocGen</masterLabel>
</DocumentGenerationSetting>
```

The following is an example package.xml that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2021/10/metadata">
    <types>
        <members>DocumentGeneration</members>
        <name>Settings</name>
    </types>
    <version>53.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type doesn't support the wildcard character * (asterisk) in the package.xml manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

DynamicFormsSettings

Represents the settings related to Dynamic Forms.

Parent Type and Manifest Access

This type extends the Metadata metadata type and inherits its fullName field.

In the package manifest, all the settings metadata types for the org are accessed using the "Settings" name. See Settings for more details.

File Suffix and Directory Location

`DynamicFormsSettings` values are stored in the `DynamicForms.settings` file in the `settings` folder. The `.settings` files are different from other named components, because there is only one settings file for each settings component.

Version

DynamicFormsSettings components are available in API version 58.0 and later.

Special Access Rules

There are no additional access requirements that are specific to this type.

Fields

Field Name	Description
`enableFormsOnMobile`	**Field Type** boolean **Description** Determines whether an org has Dynamic Forms for Mobile (Beta) enabled (`true`) or not (`false`). Note: This feature is a Beta Service. Customer may opt to try such Beta Service in its sole discretion. Any use of the Beta Service is subject to the applicable Beta Services Terms provided at Agreements and Terms.

Declarative Metadata Sample Definition

The following is an example of a DynamicFormsSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<DynamicFormsSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <enableFormsOnMobile>true</enableFormsOnMobile>
</DynamicFormsSettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>DynamicForms</members>
        <name>Settings</name>
    </types>
    <version>58.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

EACSettings

Represents the Einstein Activity Capture metadata type. Use Einstein Activity Capture to add emails and events from your Microsoft or Google account to the activity timeline of related Salesforce records. Automatically sync contact and event data between your Microsoft or Google account and Salesforce. This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

`EACSettings` components have the suffix `EAC` and are stored in the `Settings` folder.

Version

`EACSettings` components are available in API version 48.0 and later.

Fields

Field Name	Field Type	Description
`addRcCompToFlexiPages`	boolean	Indicates whether the Recommended Connections component is automatically added (`true`) to standard contact, lead, and person account Lightning record pages or not (`false`). Default value is `false`. Available in API version 53.0 and later.
`autoContactCreationPref`	boolean	Indicates whether Automatic Contact Creation is enabled (`true`) or not (`false`). Default value is `false`. Available in API version 61.0 and later.
`autoContactEnrichmentPref`	boolean	Indicates whether Automatic Contact Enhancements is enabled (`true`) or not (`false`). Default value is `false`. Available in API version 61.0 and later.
`automatedEmailFilter`	boolean	Indicates whether the setting to prevent automated emails from being shared is on (`true`) or not (`false`). Available in API version 54.0 and later.

Field Name	Field Type	Description
`autoPopulateGoogleMeetLinks`	boolean	Indicates whether the details to join with Google Meet are added (`true`) to the Google version of the event or not (`false`) when sales reps create events in Salesforce. The Google Meet details don't sync back to Salesforce. Default value is `false`. Available in API version 53.0 and later.
`enableActivityAnalyticsPref`	boolean	Indicates whether the Activities dashboard is enabled (`true`) or not (`false`). For orgs that enable Einstein Activity Capture after the Summer '21 release, the default value is `false`. Available in API version 53.0 and later.
`enableActivityCapture`	boolean	Indicates whether Einstein Activity Capture is enabled (`true`) or not (`false`). `provisionProductivityFeatures` must be `true` to use this feature. To ensure that your org's requirements for handling sensitive data are met, see Einstein Activity Capture Considerations. Default value is `false`.
`enableActivityMetrics`	boolean	Indicates whether Activity Metrics are enabled (`true`) or not (`false`). `enableActivityCapture` must be `true` to use this feature. Before enabling this feature, see Considerations for Using Activity Metrics. Default value is `false`.
`enableActivitySyncEngine`	boolean	Indicates whether combined sync and capture is enabled for events, contacts, and emails (`true`) or not (`false`). Default value is `false`.
`enableEACForEveryonePref`	boolean	Indicates whether users who have the `enableActivityCapture` set to `false` can still see emails and events in their Salesforce timeline (`true`) or not (`false`). Default value is `true`.
`enableEnforceEacSharingPref`	boolean	Indicates whether new Einstein Activity Capture users are required to keep their activity sharing setting as Don't Share (`true`) or not (`false`). Users can still share individual emails and events, and respond to sharing requests from other users.
`enableInboxActivitySharing`	boolean	Indicates whether the default activity sharing for new users is set to `Everyone` (`true`) or not (`false`). For example, if `enableInboxActivitySharing` is `true`, then new Einstein Activity Capture users have their activity sharing set to `Everyone` by default. This setting does not affect the activity sharing setting of existing users.

Field Name	Field Type	Description
		Default value is `true`.
`enableInsightsInTimeline`	boolean	Indicates whether Email Insights is enabled (`true`) or not (`false`). Default value is `true`.
`enableInsightsInTimelineEacStd`	boolean	Indicates whether Email Insights is enabled for users with an Einstein Activity Capture Standard permission set (`true`) or not (`false`). For more information, see "Turn On Einstein Email Insights" in Salesforce Help. Default value is `false`.
`provisionProductivityFeatures`	boolean	Indicates whether your org is ready for productivity features to be enabled (`true`) or not (`false`). Default value is `false`.
`relationshipGraphPref`	boolean	Indicates whether Buyer Relationship Map is enabled (`true`) or not (`false`). Default value is `true`. Available in API version 61.0 and later.
`salesforceEventsOnlyPref`	boolean	Indicates whether the activity timeline shows only events that are Salesforce records (`true`) or not (`false`). For more information, see Guidelines for Using Events with Einstein Activity Capture Available in API version 53.0 and later.
`sensitiveEmailFilter`	boolean	Indicates whether the setting to prevent sensitive emails from being shared is on (`true`) or not (`false`). Available in API version 54.0 and later.
`syncInternalEvents`	boolean	Indicates whether internal events sync between the connected account and Salesforce (`true`) or not (`false`). Events are internal when all attendees are part of the internal domain. Available in API version 53.0 and later.
`s2XSvcAccEmail`	boolean	Indicates whether a warning email has be sent to the customer if they're using Service Account OAuth on Exchange Online (`true`) or not (`false`). Default value is `false`. Available in API version 62.0 and later.

Declarative Metadata Sample Definition

The following is an example of the EAC.settings file:

```
<?xml version="1.0" encoding="UTF-8"?>
<EACSettings xmlns="http://soap.sforce.com/2006/04/metadata">
 <enableActivityCapture>true</enableActivityCapture>
 <enableActivityMetrics>true</enableActivityMetrics>
 <enableActivitySyncEngine>true</enableActivitySyncEngine>
 <enableEACForEveryonePref>true</enableEACForEveryonePref>
 <enableInboxActivitySharing>true</enableInboxActivitySharing>
 <enableInsightsInTimeline>true</enableInsightsInTimeline>
 <enableInsightsInTimelineEacStd>true</enableInsightsInTimelineEacStd>
 <provisionProductivityFeatures>true</provisionProductivityFeatures>
</EACSettings>
```

Example Package Manifest

The following is an example package manifest used to deploy or retrieve the EAC settings metadata:

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
 <types>
  <members>EAC</members>
  <name>Settings</name>
 </types>
 <version>29.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

EinsteinAgentSettings

Represents settings for Einstein classification apps, Einstein Case Classification and Einstein Case Wrap-Up, in an org. This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

EinsteinAgentSettings values are stored in the `EinsteinAgent.settings` file in the `settings` folder. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

EinsteinAgentSettings components are available in API version 52.0 and later. In API version 52.0, we renamed CaseClassificationSettings components to EinsteinAgentSettings components to reflect how we consolidated settings for Einstein Case Classification and Einstein Case Wrap-Up. CaseClassificationSettings components are available in API version 47.0 through 51.0.

Fields

Field Name	Field Type	Description
`einsteinAgentRecommendations`	boolean	Indicates whether Einstein classification apps are enabled in your org. The default value is `false`.
`reRunAttributeBasedRules`	boolean	If `true`, skills-based routing rules are run after Einstein Case Classification automatically updates field values. The default value is `false`.
`runAssignmentRules`	boolean	If `true`, assignment rules are run after Einstein Case Classification automatically updates field values. The default value is `false`.

Declarative Metadata Sample Definition

The following is an example of a EinsteinAgentSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<EinsteinAgentSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <einsteinAgentRecommendations>true</einsteinAgentRecommendations>
    <reRunAttributeBasedRules>true</reRunAttributeBasedRules>
    <runAssignmentRules>true</runAssignmentRules>
</EinsteinAgentSettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>EinsteinAgent</members>
        <name>Settings</name>
    </types>
    <version>52.0</version>
</Package>
```

EinsteinGptSettings

Represents settings for Einstein Generative AI features in an org. This type extends the Metadata metadata type and inherits its `fullName` field

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

EinsteinGptSettings values are stored in the `EinsteinGpt.settings` file in the `settings` folder. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

EinsteinGptSettings components are available in API version 61.0 and later.

Fields

Field Name	Field Type	Description
enableEinsteinGptGlobalLangSupport	boolean	Indicates whether to turn on global languages with prompt templates. The default value is `false`.
enableEinsteinGptPlatform	boolean	Indicates whether to turn on generative AI features across Salesforce. The default value is `false`.

Declarative Metadata Sample Definition

The following is an example of an EinsteinGptSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<EinsteinGptSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <enableEinsteinGptGlobalLangSupport>true</einsteinAgentRecommendations>
    <enableEinsteinGptPlatform>true</reRunAttributeBasedRules>
</EinsteinGptSettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>EinsteinGpt</members>
        <name>Settings</name>
    </types>
    <version>63.0</version>
</Package>
```

EmailAdministrationSettings

Represents an organization's email administration settings, including email deliverability, security compliance, relay configurations, and system notifications. This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

EmailAdministrationSettings values are stored in the `EmailAdminstration.settings` file in the `settings` directory. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

EmailAdministrationSettings is available in API version 47.0 and later.

Fields

Field Name	Field Type	Description
`enableComplianceBcc`	boolean	Indicates whether a copy of each outbound email message is sent to an email address you specify (`true`) or not (`false`). This field has a default value of `false`. Note: To use this feature, you must specify an email address in **Compliance BCC Email** in Setup.
`enableEmailConsentManagement`	boolean	Indicates whether Enforce Email Privacy Settings is enabled `(true)` or not `(false)`. When enabled, Salesforce respects each recipient's email privacy preferences. Default value is `false`.
`enableEmailSenderIdCompliance`	boolean	Indicates whether outgoing emails comply with Sender ID email protocols (`true`) or not (`false`). This field has a default value of `false`. To enable this preference, `enableEmailSpfCompliance` must be set to `true`. Note: Evaluate the multiple standard email security protocols (SPF, DKIM, and DMARC) supported by Salesforce before you enable this setting.
`enableEmailSpfCompliance`	boolean	Indicates whether outgoing emails comply with Sender Policy Framework (SPF) email authentication (`true`) or not (`false`). This field has a default value of `true`.
`enableEmailToSalesforce`	boolean	Indicates whether Email to Salesforce is enabled (`true`) or disabled (`false`). This field has a default value of `false`.
`enableEmailWorkflowApproval`	boolean	Indicates whether users can respond to email approval requests directly from their email (`true`) or not (`false`). This field has a default value of `false`. See Considerations for Email Approval before enabling this field.
`enableEnhancedEmailEnabled`	boolean	Indicates whether Enhanced Email is enabled `(true)` or not `(false)`. Default value is `true`.
`enableHandleBouncedEmails`	boolean	Indicates whether emails sent from Salesforce to an invalid email address bounce back to Salesforce (`true`) or not (`false`). This field has a default value of `true`. With bounce handling enabled, reps know which lead, contact, or person account has a bad email address, and they know which specific email wasn't delivered.
`enableHtmlEmail`	boolean	Indicates whether users receive Email-To-Case emails in HTML format (`true`) or receive a text version instead (`false`). This field has a default value of `false`. When this field is set to `true`, users receive a warning message

Field Name	Field Type	Description
		about potential malicious HTML before they view incoming HTML email content.
`enableInternationalEmailAddresses`	boolean	Indicates whether non-Latin-based characters are allowed in email addresses (`true`) or not (`false`) when sending emails to and from Salesforce. This field has a default value of `true` in orgs created in Summer '20 or later. In orgs created in Spring '20 or earlier, the default value is `false`. Available in API version 49.0 and later. Note: Review the Email Address Internationalization prerequisites and considerations in Salesforce Help before enabling this setting.
`enableListEmailLogActivities`	boolean	Indicates whether Salesforce logs sent list emails as activities `(true)` or not `(false)`. Default value is `true`.
`enableResendBouncedEmails`	boolean	Indicates whether the system forwards a copy of each bounced email message to the sender (`true`) or only displays the bounce alert (`false`). This field has a default value of `false`. To enable this preference, `enableHandleBouncedEmails` must be set to `true`.
`enableRestrictTlsToDomains`	boolean	Indicates whether the selected Transport Layer Security (TLS) setting applies only to specific domains (`true`) or applies to all domains (`false`). This field has a default value of `false`. Note: To enable this preference, you must specify a **TLS Setting** other than Preferred and provide the comma-separated list of domains through **Deliverability** in Setup. When this field is set to `true`, any domains not in the list use the system default TLS Setting of Preferred.
`enableSendThroughGmailPref`	boolean	Deprecated.
`enableSendViaExchangePref`	boolean	Indicates whether users can use Office 365 to send emails `(true)` or not `(false)`. Default value is `false`.
`enableSendViaGmailPref`	boolean	Indicates whether users can use Gmail to send emails `(true)` or not `(false)`. Default value is `false`.
`enableUseOrgFootersForExtTrans`	boolean	Indicates whether emails sent through external email services (such as Gmail or Office 365) include the Salesforce footer (`true`) or not (`false`). This field has a default value of `false`.

Field Name	Field Type	Description
`sendMassEmailNotification`	boolean	Indicates whether users receive an auto-generated status email from Salesforce for each mass email they send (`true`) or not (`false`). This field has a default value of `true`.
`sendTextOnlySystemEmails`	boolean	Indicates whether all system emails are sent via text only (`true`) or allow other formats (`false`). This field has a default value of `false`.

Declarative Metadata Sample Definition

The following is an example of an EmailAdministrationSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
              <EmailAdministrationSettings xmlns="http://soap.sforce.com/2006/04/metadata">

                <enableEmailWorkflowApproval>false</enableEmailWorkflowApproval>
                <enableComplianceBcc>false</enableComplianceBcc>
                <enableEmailSenderIdCompliance>false</enableEmailSenderIdCompliance>
                <enableEmailSpfCompliance>true</enableEmailSpfCompliance>
                <enableEmailToSalesforce>false</enableEmailToSalesforce>
                <enableHandleBouncedEmails>true</enableHandleBouncedEmails>
                <enableHtmlEmail>true</enableHtmlEmail>
              <enableInternationalEmailAddresses>true</enableInternationalEmailAddresses>

                <enableResendBouncedEmails>false</enableResendBouncedEmails>
                <enableRestrictTlsToDomains>false</enableRestrictTlsToDomains>
                <sendMassEmailNotification>true</sendMassEmailNotification>
                <sendTextOnlySystemEmails>false</sendTextOnlySystemEmails>
                <enableUseOrgFootersForExtTrans>false</enableUseOrgFootersForExtTrans>
                <enableSendViaGmailPref>false</enableSendViaGmailPref>
                <enableSendViaExchangePref>true</enableSendViaExchangePref>
                <enableListEmailLogActivities>false</enableListEmailLogActivities>
                <enableEnhancedEmailEnabled>false</enableEnhancedEmailEnabled>
                <enableEmailConsentManagement>false</enableEmailConsentManagement>
                </EmailAdministrationSettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
                <Package xmlns="http://soap.sforce.com/2006/04/metadata">
                <types>
                <members>EmailAdministration</members>
                <name>Settings</name>
                </types>
                <version>49.0</version>
                </Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SEE ALSO:

Salesforce Help: Email Address Internationalization

EmailIntegrationSettings

Represents an org's settings for the Outlook integration, Gmail integration, and Salesforce Inbox. This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

EmailIntegrationSettings values are stored in the `EmailIntegration.settings` file in the `settings` directory. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

EmailIntegrationSettings fields are available in API version 47.0 and later.

Fields

Field Name	Field Type	Description
`doesEmailLogAsEmailMessageInOutlook`	boolean	Indicates whether the Outlook integration logs emails to the Email Message object (`true`) or as tasks (`false`). The default value is `true`. This field can only be used if the `enableOutlookIntegration` field is set to `true`.
`doesGmailStayConnectedToSalesforce`	boolean	Indicates whether Gmail integration users log in to Salesforce from Gmail each time their session expires. If set to `true`, users log in from the Gmail integration one time, and their credentials are remembered the next time they use the Gmail integration. If set to `false`, users log in to Salesforce from the Gmail integration each time their Salesforce session expires. The default value is `false`. This field can only be used if the `enableGmailIntegration` field is set to `true`.
`enableContactAndEventSync`	boolean	Indicates whether users can sync calendar events and contacts between Salesforce and their Microsoft and Google accounts (`true`) or not (`false`). Lightning Sync or Einstein Activity Capture must be enabled to use this feature. The default value is `false`.

Field Name	Field Type	Description
		Available in API version 48.0 and later.
`enableContextualEverywhere`	boolean	Indicates whether contextual insights in Sales Cloud Everywhere, available in the Salesforce Chrome extension, are enabled (`true`) or not (`false`). This field is available in API version 58.0 and later.
`enableEmailTrackingInMobile`	boolean	Indicates whether Salesforce Inbox users can track emails (`true`) or not (`false`) while in the Outlook integration with Inbox or the Gmail integration with Inbox. It also controls email tracking in the Inbox mobile app and legacy versions of Inbox. The default value is `true`.
`enableEngageForOutlook`	boolean	Indicates whether Engage For Outlook is enabled (`true`) or not (`false`). When set to `true`, Engage users can connect their Outlook account and send Engage emails from their Outlook inbox.
`enableExtensionHostUnrestricted`	boolean	Indicates whether the contextual web experience is enabled in the chrome extension or not.
`enableGmailIntegration`	boolean	Indicates whether the Gmail integration is enabled (`true`) or not (`false`). When set to `true`, G Suite users with the Gmail integration can connect their Gmail account and work with Salesforce data in their email. The default value is `true`.
`enableInboxInternalEmailTracking`	boolean	Indicates whether a read-receipt status is shown for emails that are sent within the same domain (`true`) or not (`false`). This field is available in API version 58.0 and later.
`enableInboxMobileIntune`	boolean	Indicates whether Inbox is enabled to use Microsoft Intune to manage security settings (`true`) or not (`false`). When set to `true`, Inbox mobile users need a Microsoft Intune license to log in to the app. The default value is `false`. Available in API version 50.0 and later.
`enableOutlookIntegration`	boolean	Indicates whether the Outlook integration is enabled (`true`) or not (`false`). When set to `true`, Outlook users with the Outlook integration can connect their Outlook account and work with Salesforce data in their email. The default value is `false`.
`enableProductivityFeatures`	boolean	Indicates whether Inbox features, such as Availability and Send later, are available (`true`) or not available (`false`) in the Outlook or Gmail integration. The default value is `false`. This field can only be used if either the `enableOutlookIntegration` or `enableGmailIntegration` field is set to `true` and if the org has an Inbox license.

Field Name	Field Type	Description
		Note: To see Inbox features, users must also have either the Inbox with Einstein Activity Capture or the Inbox without Einstein Activity capture permission set.
enableSupplementalContactInfoInMobile	boolean	Indicates whether Salesforce Inbox mobile app users see third-party contact information when contacts are shown (`true`) or not (`false`) in the Inbox mobile app. The default value is `false`.
isLayoutCustomizationAllowed	boolean	Indicates whether Salesforce admins are allowed (`true`) or not allowed (`false`) to create custom email application panes using the Lightning App Builder. The email application pane defines the layout of the Salesforce pane in Outlook and Gmail. The default value is `true`. This field can only be used if either the `enableOutlookIntegration` or `enableGmailIntegration` field is set to `true`.
orgIsSyncingEventsOutbound	boolean	Indicates whether changes to Salesforce events sync to Outlook and Google calendars (`true`) or not (`false`). Note: This field is set by Salesforce. We do not recommend that you set this field manually, as doing so may cause interruptions in your org's event syncing. This field is available in API version 50.0 and later
shouldUseTrustedDomainsList	boolean	Indicates if the web domains listed in the Outlook Integration & Sync page in Salesforce Setup are supported (`true`) or not (`false`). These domains are for users who access email using Outlook on the web. The default value is `true`. This field can only be used if the `enableOutlookIntegration` field is set to `true`.

Declarative Metadata Sample Definition

The following is an example of a EmailIntegrationSettings file.

```
<?xml version="1.0" encoding="UTF-8"?>
<EmailIntegrationSettings xmlns="http://soap.sforce.com/2006/04/metadata">
        <enableContactAndEventSync>true</enableContactAndEventSync>
        <enableProductivityFeatures>true</enableProductivityFeatures>
        <doesGmailStayConnectedToSalesforce>true</doesGmailStayConnectedToSalesforce>
        <enableOutlookIntegration>true</enableOutlookIntegration>
        <enableGmailIntegration>true</enableGmailIntegration>
        <isLayoutCustomizationAllowed>true</isLayoutCustomizationAllowed>
        <doesEmailLogAsEmailMessageInOutlook>false</doesEmailLogAsEmailMessageInOutlook>
        <shouldUseTrustedDomainsList>false</shouldUseTrustedDomainsList>
        <enableEmailTrackingInMobile>true</enableEmailTrackingInMobile>
       <enableSupplementalContactInfoInMobile>false</enableSupplementalContactInfoInMobile>
```

```
        <enableEngageForOutlook>true</enableEngageForOutlook>
</EmailIntegrationSettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>EmailIntegration</members>
        <name>Settings</name>
    </types>
    <version>47.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

EmailTemplateSettings

Represents an org's email template settings. This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

EmailTemplateSettings values are stored in the `EmailTemplate.settings` file in the `settings` directory. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

EmailTemplateSettings is available in API version 47.0 and later.

Fields

Field Name	Field Type	Description
`enableTemplateEnhancedFolderPref`	boolean	Indicates whether Folders and Enhanced Sharing for Email Templates is enabled (`true`) or not (`false`). This feature allows users to create and manage folders for email templates. Default value is `false`.

Declarative Metadata Sample Definition

The following is an example of the package file.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>EmailTemplate</members>
        <name>Settings</name>
    </types>
</Package>
```

The package file references the following EmaillTemplate.settings file.

```
<?xml version="1.0" encoding="UTF-8"?>
<EmailTemplateSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <enableTemplateEnhancedFolderPref>true</enableTemplateEnhancedFolderPref>
</EmailTemplateSettings>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

EmployeeUserSettings

Represents the employee-user settings used for automatically creating or syncing employee and user data in work.com orgs. This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

A EmployeeUserSettings component file has the suffix `.settings` and is stored in the `settings` directory. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

EmployeeUserSettings components are available in API version 50.0 and later.

Special Access Rules

Access to the EmployeeUserSettings type requires the Workplace Command Center permission set license available in the org and assigned to the user. The WorkplaceCommandCenterUser scratch org feature must also be enabled in the org.

Fields

Field Name	Field Type	Description
emailEncoding	string	Required. The default encoding setting is Unicode: UTF-8. Valid values include: • UTF-8—Unicode (UTF-8) • ISO-8859-1—General US & Western Europe (ISO-8859–1, ISO-LATIN-1) • Shift_JIS—Japanese (Shift-JIS) • ISO-2022-JP—Japanese (JIS) • EUC-JP—Japanese (EUC-JP) • x-SJIS_0213—Japanese (Shift-JIS_2004) • ks_c_5601-1987—Korean (ks_c_5601-1987) • Big5—Traditional Chinese (Big5) • GB2312—Simplified Chinese (GB2312) • Big5-HKSCS—Traditional Chinese Hong Kong (Big5–HKSCS)
enableEmployeeAutoCreateUser	boolean	If true, users are auto-created when a new employee record is created. The default value for this field is false.
enableEmployeeIsSourceOfTruth	boolean	If true, the employee record is the source of truth. The default value for this field is false.
permset	string	Represents a set of permissions that's used to grant more access to a user. You can use permission sets to grant access but not to deny access.
profile	string	Required. Represents a user profile. A profile defines a user's permission to perform different functions within Salesforce.
usernameSuffix	string	Represents a domain name. We create a unique login by combining this domain name with each employee's username.

Declarative Metadata Sample Definition

The following is an example EmployeeUser.settings-meta.xml that deploys the EmployeeUserSettings metadata to an org. The file is in the dir path force-app/main/default/settings:

```
<?xml version="1.0" encoding="UTF-8"?>
<EmployeeUserSettings xmlns="http://soap.sforce.com/2006/04/metadata">
<emailEncoding>ISO-8859-1</emailEncoding>
<enableEmployeeAutoCreateUser>true</enableEmployeeAutoCreateUser>
<enableEmployeeIsSourceOfTruth>false</enableEmployeeIsSourceOfTruth>
<profile>MarketingProfile</profile>
<usernameSuffix>example.com</usernameSuffix>
</EmployeeUserSettings>
```

The following example of `package.xml` file retrieves the EmployeeUserSettings metadata:

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
<types>
<name>Settings</name>
<members>EmployeeUser</members>
</types>
<version>50.0</version>
</Package>
```

EnhancedNotesSettings

Represents an org's enhanced note settings, such as enabling enhanced notes and enabling tasks in enhanced notes.This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

EnhancedNotesSettings values are stored in the `EnhancedNotes.settings` file in the `settings` directory.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

Version

EnhancedNotesSettings is available in API version 47.0 and later.

Fields

Field Name	Field Type	Description
`enableEnhancedNotes`	boolean	Indicates whether enhanced notes are enabled `(true)` or not `(false)`. With enhanced notes, users can relate a note to multiple records, access version history, and enjoy enhanced format options. Users must have the Use New Notes permission to use enhanced notes. Default value is `true`.
`enableTasksOnEnhancedNotes`	boolean	Indicates whether users can create tasks from notes `(true)` or not `(false)`. In the Salesforce app, users can create a task from a note by swiping a line on the note. Alternatively, they can tap in the toolbar to add or update the status of an action item. Users must have the Use New Notes permission to use enhanced notes. Default value is `true`.

Declarative Metadata Sample Definition

The following is an example of the `EnhancedNotesSettings.settings` file:

```
<?xml version="1.0" encoding="UTF-8"?>
<EnhancedNotesSettings xmlns="http://soap.sforce.com/2006/04/metadata">
 <enableEnhancedNotes>true</enableEnhancedNotes>
 <enableTasksOnEnhancedNotes>true</enableTasksOnEnhancedNotes>
</EnhancedNotesSettings>
```

Example Package Manifest

The following is an example package manifest used to deploy or retrieve the EnhancedNotesSettings metadata:

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
 <types>
  <members>EnhancedNotes</members>
  <name>Settings</name>
 </types>
 <version>47.0</version>
</Package>
```

EncryptionKeySettings

Represents an org's encryption key settings, such as customer-supplied keys options and key derivation settings. This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

EncryptionKeySettings values are stored in the `EncryptionKey.settings` file in the `settings` folder. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

EncryptionKeySettings is available in API versions 47.0 and later.

Special Access Rules

To enable EncryptionKeySettings, you need the Customize Application and Manage Encryption Keys permissions.

Fields

Field Name	Field Type	Description
`canEncryptTransactionalDatabase`	boolean	Indicates whether users can encrypt the transactional database (`true`) or can't (`false`). The default value is `false`. If set to `true`, transactional database encryption is enabled and a transactional

Field Name	Field Type	Description
		database tenant secret is generated. Available in API version 63.0 and later.
`canExternalKeyManagement`	boolean	Indicates whether External Key Management is available (`true`) or not (`false`). The default value is `false`. If set to `true`, users can configure connections to external key stores and use keys saved in those key stores to encrypt and decrypt Salesforce data. Available in API version 63.0 and later.
`canManageDataCloudKeys`	boolean	Indicates whether key management for Data Cloud is enabled and an initial customer-managed root key is present (`true`) or not (`false`). Available in API version 63.0 and later.
`canOptOutOfDerivationWithBYOK`	boolean	Indicates that users can opt out of key derivation processes on a key-by-key basis when they upload key material (`true`) or can't (`false`). The default value is `false`.
`enableCacheOnlyKeys`	boolean	Indicates whether the Cache-Only Key Service is available (`true`) or not (`false`). The default value is `false`. If set to `true`, users can configure a cache-only key callout connection and apply key material stored outside of Salesforce to data on demand.
`enableReplayDetection`	boolean	Indicates whether cache-only key callouts are protected from replay attacks by a nonce (`true`) or not (`false`). Requires `enableCacheOnlyKeys="true"` before setting `enableReplayDetection` to `true`.

Declarative Metadata Sample Definition

The following is an example of the EncryptionKey.settings file:

```
<?xml version="1.0" encoding="UTF-8"?>
<EncryptionKeySettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <canEncryptTransactionalDatabase>true</canEncryptTransactionalDatabase>
    <canManageDataCloudKeys>true</canManageDataCloudKeys>
    <enableExternalKeyManagement>true</enableExternalKeyManagement>
</EncryptionKeySettings>
```

Example Package Manifest

The following is an example package manifest used to deploy or retrieve the encryption key settings metadata for an organization:

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>EncryptionKey</members>
        <name>Settings</name>
    </types>
    <version>47.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

EntitlementSettings

Represents an organization's entitlement settings.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

EntitlementSettings values are stored in the `Entitlements.settings` file in the `settings` directory. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

EntitlementSettings is available in API version 27.0 and later.

Fields

Field Name	Field Type	Description
`assetLookupLimitedToActiveEntitlementsOnAccount`	boolean	Indicates whether entitlements-related lookup filters on cases return only the assets related to the active entitlements on the case's account (`true`) or not (`false`).
`assetLookupLimitedToActiveEntitlementsOnContact`	boolean	Indicates whether entitlements-related lookup filters on cases return only the assets related to the active entitlements on the case's contact (`true`) or not (`false`).
`assetLookupLimitedToSameAccount`	boolean	Indicates whether entitlements-related lookup filters on cases return only the assets related to the case's account (`true`) or not (`false`).
`assetLookupLimitedToSameContact`	boolean	Indicates whether entitlements-related lookup filters on cases return only the assets related to the case's contact (`true`) or not (`false`).
`enableEntitlements`	boolean	Indicates whether entitlements are enabled (`true`) or not (`false`).
`enableEntitlementVersioning`	boolean	Indicates whether entitlement versioning is enabled (`true`) or not (`false`). This field is available in API version 28.0 and later.
`enableMilestoneFeedItem`	boolean	When set to `true`, indicates whether to post to the feed and the record owner's profile page when a milestone is

Field Name	Field Type	Description
		completed or violated. When set to `false`, indicates that no post occurs when a milestone is completed or violated. This field is available in API version 47.0 and later.
`enableMilestoneStoppedTime`	boolean	Indicates whether to show the **Stopped Time** and **Actual Elapsed Time** fields on an entitlement milestone (`true`) or not (`false`). This field is available in API version 47.0 and later.
`entitlementLookupLimitedToActiveStatus`	boolean	Indicates whether entitlements-related lookup filters on cases return only active entitlements (`true`) or not (`false`).
`entitlementLookupLimitedToSameAccount`	boolean	Indicates whether entitlements-related lookup filters on cases return only the entitlements related to the case's account (`true`) or not (`false`).
`entitlementLookupLimitedToSameAsset`	boolean	Indicates whether entitlements-related lookup filters on cases return only the entitlements related to the case's asset (`true`) or not (`false`).
`entitlementLookupLimitedToSameContact`	boolean	Indicates whether entitlements-related lookup filters on cases return only the entitlements related to the case's contact (`true`) or not (`false`).
`ignoreMilestoneBusinessHours`	boolean	Indicates whether to show the time remaining on an event milestone in actual hours (`true`) or business hours (`false`). This field is available in API version 47.0 and later.

Declarative Metadata Sample Definition

This is a sample entitlements settings file.

```
<?xml version="1.0" encoding="UTF-8"?>
<EntitlementSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <assetLookupLimitedToActiveEntitlementsOnAccount>
        false
    </assetLookupLimitedToActiveEntitlementsOnAccount>
    <assetLookupLimitedToActiveEntitlementsOnContact>
        false
    </assetLookupLimitedToActiveEntitlementsOnContact>
    <assetLookupLimitedToSameAccount>
        false
    </assetLookupLimitedToSameAccount>
    <assetLookupLimitedToSameContact>
        false
    </assetLookupLimitedToSameContact>
    <enableEntitlements>
```

```
        true
    </enableEntitlements>
    <entitlementLookupLimitedToActiveStatus>
        false
    </entitlementLookupLimitedToActiveStatus>
    <entitlementLookupLimitedToSameAccount>
        false
    </entitlementLookupLimitedToSameAccount>
    <entitlementLookupLimitedToSameAsset>
        false
    </entitlementLookupLimitedToSameAsset>
    <entitlementLookupLimitedToSameContact>
        false
    </entitlementLookupLimitedToSameContact>
</EntitlementSettings>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

EventSettings

Represents an org's platform event settings for Event Monitoring.

File Suffix and Directory Location

EventSettings components have the suffix `.settings` and are stored in the `settings` folder.

Version

EventSettings components are available in API version 47.0 and later.

Fields

Field Name	Field Type	Description
`bypassMeteringBlock`	boolean	Determines the behavior of metering service protection for Transaction Security policies. When `true`, metering occurs but doesn't block the user operation that triggered the policy. When `false` and a policy can't be handled within three seconds, metering occurs and the user's action is blocked. Default value is `false`.
`enableDeleteMonitoringData`	boolean	Allows (`true`) or disallows (`false`) users to delete event log files and LoginEvent data. Users require the Delete Event Monitoring Records user permission, which is available when this setting is enabled. Default value is `false`.

Field Name	Field Type	Description
`enableDynamicStreamingChannel`	boolean	Enables (`true`) or disables (`false`) the dynamic creation of a streaming channel when you subscribe to generic streaming. Default value is `false`.
`enableEventLogGeneration`	boolean	Enables (`true`) or disables (`false`) the generation of event monitoring log files. Default value is `false`.
`enableEventLogObjectFeature`	boolean	Enables (`true`) or disables (`false`) event log objects. Default value is `false`.
`enableEventLogWaveIntegration`	boolean	Enables (`true`) or disables (`false`) the integration of event monitoring log files and Analytics apps. Analytics apps help you visualize your user's activity. Default value is `false`.
`enableLightningLoggerEvents`	boolean	Enables (`true`) or disables (`false`) the generation of Lightning Logger events. Lightning Logger events track information about custom Lightning web components. Default value is `false`. Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.
`enableLoginForensics`	boolean	Enables (`true`) or disables (`false`) the Login Forensics feature. Login Forensics helps you track and audit your org's user login activity. Default value is `false`. Available in API versions 47.0–49.0. Tip: In versions 50.0 and later, enable LoginEvent on the Event Manager Setup page.
`enableStreamingApi`	boolean	Enables (`true`) or disables (`false`) Streaming API in the org. Default value is `true`.
`enableTerminateOldestSession`	boolean	Determines the behavior of legacy transaction security policies that trigger an end-session action during an API-based login (a login that doesn't come through the UI.) An end-session action occurs when a user exceeds the maximum number of allowed Salesforce sessions. When `true`, and a user triggers an end-session action, Salesforce terminates the user's oldest session until the user is in compliance. When set to `false`, Salesforce blocks the most recent user's attempt to log in and doesn't allow a new user session. Default value is `false`. Available in API versions 47.0–49.0. Note: As of Summer '20, Legacy Transaction Security is a retired feature in all Salesforce orgs.
`enableTransactionSecurityPolicies`	boolean	Enables (`true`) or disables (`false`) the ability to create and use transaction security policies in the Salesforce UI. Default value is `false`.

Field Name	Field Type	Description
`enableApexLimitEvents`	boolean	Enables (`true`) or disables (`false`) the Apex Limit Events (Pilot) feature. Default value is `false`. Note: The Apex Limit Events (Pilot) feature has been discontinued so don't use this field.
`eventLogRetentionDuration`	integer	Specifies the number of days (between 30 and 365) that your event log file data is retained for. If this value is not set, your event log file data is retained for your org's default retention period.

Declarative Metadata Sample Definition

The following is an example of an `EventSettings.settings` file.

```
<?xml version="1.0" encoding="UTF-8"?>
<EventSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <enableDeleteMonitoringData>true</enableDeleteMonitoringData>
    <enableDynamicStreamingChannel>false</enableDynamicStreamingChannel>
    <enableEventLogWaveIntegration>true</enableEventLogWaveIntegration>
    <enableStreamingApi>true</enableStreamingApi>
    <enableTransactionSecurityPolicies>true</enableTransactionSecurityPolicies>
</EventSettings>
```

Example Package Manifest

The following is an example package manifest used to deploy or retrieve the Event settings metadata:

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
   <types>
       <members>Event</members>
       <name>Settings</name>
   </types>
   <version>47.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ExperienceBundleSettings

Represents the org setting that enables the ExperienceBundle metadata type for Aura sites in Experience Cloud. The setting doesn't affect LWR sites, which use ExperienceBundle by default. This type extends the Metadata metadata type and inherits its `fullName` field.

Note: ExperienceBundle is a text-based code structure of the settings and site components, such as pages, branding sets, and themes, that make up an Experience Builder site. Developers can quickly update and deploy one or more Experience Builder sites programmatically using their preferred development tools.

File Suffix and Directory Location

ExperienceBundleSettings values are stored in a single file named `ExperienceBundle.settings` in the `settings` directory. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

ExperienceBundleSettings is available in API version 46.0 and later.

Fields

Field Name	Field Type	Description
`enableExperienceBundleMetadata`	boolean	Indicates whether the ExperienceBundle metadata type is enabled for Aura sites. Default is `false`. LWR sites use ExperienceBundle by default.

Declarative Metadata Sample Definition

Here's an example of `ExperienceBundle.settings` that references the previous definition.

```
<ExperienceBundleSettings xmlns="http://soap.sforce.com/2006/04/metadata">
        <enableExperienceBundleMetadata>true</enableExperienceBundleMetadata>
</ExperienceBundleSettings>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SEE ALSO:

ExperienceBundle

ExternalClientAppSettings

Represents settings to enable the External Client App feature and provide access to the OAuth consumer secret.

Parent Type and Manifest Access

This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all the settings metadata types are accessed using the Settings name. See Settings for more details.

File Suffix and Directory Location

`ExternalClientAppSettings` values are stored in the `.settings` file in the `settings` folder. The `.settings` files are different from other named components, because there's only one settings file for each settings component.

Version

ExternalClientAppSettings components are available in API version 58.0 and later.

Fields

Field Name	Field Type	Description
`enableClientSecretInRestApiAccess`	boolean	Indicates whether OAuth consumer secrets can be accessed through the `credentials` REST API (`true`) or not (`false`). This field is available in API version 62.0 and later.
`enableConsumerSecretApiAccess`	boolean	Indicates whether OAuth consumer secrets can be accessed through Metadata API (`true`) or not (`false`).
`enablePackageEcaOauthFromDevOrg`	boolean	Indicates whether you can package an external client app that is owned by an org, that isn't a Dev Hub org, (`true`) or not (`false`). It's recommended that you only package OAuth-enabled external client apps that are owned by a Dev Hub org. This practice maintains the integrity of the OAuth client credentials by storing them on an org that won't be deleted. If you have other ways to maintain the OAuth client credentials, you can enable this field and package an external client app form a different type of org. External client apps built on ephemeral orgs, like Scratch orgs or sandboxes, can't be packaged even with this field set to true. This field is available in API version 62.0 and later.

Declarative Metadata Sample Definition

The example shows a settings file component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ExternalClientAppSettings xmlns="http://soap.sforce.com/2006/04/metadata">
        <enableClientSecretInRestApiAccess>true</enableClientSecretInRestApiAccess>
        <enableConsumerSecretApiAccess>true</enableConsumerSecretApiAccess>
        <enablePackageEcaOauthFromDevOrg>false</enablePackageEcaOauthFromDevOrg>
</ExternalClientAppSettings>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ExternalServicesSettings

Represents settings for an External Services registration.

Parent Type and Manifest Access

This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all the settings metadata types for the org are accessed using the "Settings" name. See Settings for more details.

File Suffix and Directory Location

`ExternalServicesSettings` values are stored in the `externalServicesSettings.settings` file in the `settings` folder. The `.settings` files are different from other named components, because there is only one settings file for each settings component.

Version

ExternalServicesSettings components are available in API version 47.0 to 55.0. This metadata type is removed in version 56.0 and later.

As of Winter '23, External Services automatically validates the schema and provides detailed messages for any errors in the UI at registration time so that you can make corrections. See Register an External Service in Salesforce Help.

Fields

Field Name	Description
`enableIgnoreUnsupportedOperations`	**Field Type** boolean **Description** Indicates whether your registration should ignore and filter out unsupported schema operations rather than fail the entire registration (`true`), or whether a specification with invalid schema operations can't be registered successfully (`false`). Detailed schema errors pertaining to unsupported operations are shown only if this flag is `false`.

Declarative Metadata Sample Definition

The following is an example of an ExternalServicesSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ExternalServicesSettings xmlns="http://soap.sforce.com/2006/04/metadata">
  <enableIgnoreUnsupportedOperations>true</enableIgnoreUnsupportedOperations>
</ExternalServicesSettings>
```

The following is an example `package.xml` that references the previous definition.

```
<types>
    <members>ExternalServices</members>
```

```
    <name>Settings</name>
</types>
```

Wildcard Support in the Manifest File

This metadata type doesn't support the wildcard character * (asterisk) in the package.xml manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

FieldServiceSettings

Represents an organization's Field Service settings.

To learn more about Field Service settings, see Enable Field Service in Salesforce Help.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for more details.

Version

FieldServiceSettings is available in API version 40.0 and later.

Fields

Field	Field Type	Description
apptAssistantExpiration	int	The expiration time of apptAssistantInfoUrl after which the customer stops seeing the mobile worker's location and estimated time of arrival. Available in API version 50.0 and later.
apptAssistantInfoUrl	string	The tracking URL that helps a customer see the mobile worker's estimated time of arrival and tracking information. Available in API version 50.0 and later.
apptAssistantRadiusUnitValue	ApptAssistantRadiusUnit (enumeration of type string)	The units for specifying the geofence radius. When the mobile worker enters this area, a Last Mile notification is automatically sent. Valid values are: • Kilometer • Meter • Mile • Yard Available in API version 50.0 and later.
apptAssistantRadiusValue	int	The geofence radius from the service appointment's address used for sending a notification to the customer when the mobile

Field	Field Type	Description
		worker approaches the address. Available in API version 50.0 and later.
apptAssistantStatus	string	The status on the service appointment used to trigger En Route notification. The value has to match one of the service appointment's Status field options. When the mobile worker selects this status on a service appointment, the customer receives the En Route notification with tracking details. Available in API version 50.0 and later.
canPopulateGoogleAddress	boolean	Allows desktop and mobile to send geolocation and map data to Google and Apple. Available in API version 57.0 and later.
canSendAppCenterCrashReports	boolean	Allows Salesforce to send crash reports to Microsoft App Center. Available in API version 57.0 and later.
canStoreMobileAnalytics	boolean	Allows third parties to store mobile analytics. Available in API version 57.0 and later.
deepLinkPublicSecurityKey	string	Provides a public security key for users accessing a deep link action in the Field Service mobile app. Hide the redirection warning by configuring the deep link URL with a security key. The deep link URL then processes the security check. Available in API version 54.0 and later.
doesAllowEditSaForCrew	boolean	Lets service crew members edit their service appointments. This setting applies only if `doesShareSaWithAr` is selected. For assigned resources of type Crew, crew members get Read-Write access to their service appointment and, if `doesShareSaParentWoWithAr` is selected, to their service appointments' parent work orders.
doesShareSaParentWoWithAr	boolean	Shares service appointments' parent work orders with their assigned resources. This setting applies only if `doesShareSaWithAr` is selected and sharing access for work orders is set to Private or Public Read Only. Technician assigned resources get Read-Write access to their work

Field	Field Type	Description
		orders. For assigned resources of type Crew, the crew leader gets Read-Write access and crew members get Read access. If the service appointment's parent is a work order line item, assigned resources get access to the associated work order.
doesShareSaWithAr	boolean	Shares dispatched service appointments with their assigned resources. This setting applies only if sharing access for service appointments is set to Private or Public Read Only. Technician assigned resources get Read-Write access to their service appointments. For assigned resources of type Crew, the crew leader gets Read-Write access and crew members get Read access.
enableDocumentBuilder	boolean	Enables access to Document Builder feature.
enableFloatingWorkOrder	boolean	Enables floating work orders for the org. Allows users to create work orders with a floating recurrence cadence based on the previous work order's completion.
enablePopulateWorkOrderAddress	boolean	Enables address to be populated when work orders are generated from Maintenance Plan.
enableWorkOrders	boolean	Enables Work Orders for the org. This setting allows users to use the Work Order object, whether or not Field Service is enabled. When Field Service is enabled, you can't turn off Work Orders.
enableWorkPlansAutoGeneration	boolean	Allows work plans and their work steps to be generated automatically when a work order or a work order line item is newly created. The specific work plans and work steps to be generated depends on matching criteria specified in Work Plan Selection Rules. Available in API version 52.0 and later.
enableWorkStepManualStatusUpdate	boolean	Allows a work step status to be updated manually. A prompt suggests a status update that users can accept or defer.
fieldServiceNotificationsOrgPref	boolean	Turns on in-app notifications for the Salesforce mobile app and Lightning Experience users. Notifications are sent when any of the

Field	Field Type	Description
		following actions occurs on a work order or work order line item that they own or follow: • A text or file post is added • A tracked field is updated • The record owner changes • The resource assignments change on a related service appointment If the option to track all related objects is selected in the feed tracking settings for work orders, users are also notified when child records of work orders—such as service appointments—are created or deleted.
`fieldServiceOrgPref`	boolean	Indicates whether Field Service is enabled.
`isGeoCodeSyncEnabled`	boolean	Syncs the location of a Service Resource to an Inventory object.
`isLocationHistoryEnabled`	boolean	Tracks the location history of a Service Resource.
`mobileFeedbackEmails`	string	Stores an email address to which a feedback email is sent when users leave feedback from the Field Service mobile app. Available in API version 54.0 and later.
`o2EngineEnabled`	boolean	Enables Field Service Enhanced Scheduling and Optimization. The default value is false. Available in API version 55.0 and later.
`objectMappingItem`	ObjectMappingItem on page 1935	Represents an organization's custom field mapping for Work Plan or Work Step generation. Custom Fields can be mapped from WorkPlanTemplate to WorkPlan, WorkStepTemplate to WorkStep, or WorkPlanTemplateEntry to WorkStep. Available in API version 52.0 and later.
`optimizationServiceAccess`	boolean	Allows the optimization service to access data in your Salesforce org.
`serviceAppointmentsDueDateOffsetOrgValue`	int	Indicates the number of days past the Created Date that the Due Date on auto-created service appointments should fall. Work types include an option to automatically add a service appointment to new work orders or work order line items using the work type.

Field	Field Type	Description
workOrderDurationSource	WorkOrderDurationSource (enumeration of type string)	The source for the work order duration value. Possible values are: • WorkType • TotalFromWorkPlan • Custom Available in API version 55.0 and later.
workOrderLineItemSearchFields	string	The work order line item fields that the search engine should scan to suggest knowledge articles on work order line items.
workOrderSearchFields	string	The work order fields that the search engine should scan to suggest knowledge articles on work orders.

ObjectMappingItem

Represents an organization's custom field mapping for Work Plan or Work Step generation. Custom Fields can be mapped from WorkPlanTemplate to WorkPlan, WorkStepTemplate to WorkStep, or WorkPlanTemplateEntry to WorkStep. Available in API version 52.0 and later.

Field Name	Field Type	Description
mappingType	string	The type of object mapping. Valid values are: • WorkPlans_WorkPlanTemplate_WorkPlan — Maps a WorkPlanTemplate to a WorkPlan • WorkPlans_WorkStepTemplate_WorkStep — Maps a WorkStepTemplate to a WorkStep • WorkPlans_WorkPlanTemplateEntry_WorkStep — Maps a WorkPlanTemplateEntry to a WorkStep
objectMapping	ObjectMapping on page 1935	The object mapping details.

ObjectMapping

Represents a map of fields in the input object to fields in the output object.

Field Name	Field Type	Description
`inputObject`	string	Required. The name of the object type containing the source fields for mapping. Valid values are: • `WorkPlanTemplate` • `WorkStepTemplate` • `WorkPlanTemplateEntry`
`mappingFields`	[ObjectMappingField on page 1936]	Required. The mapping of source object fields to target object fields.
`outputObject`	string	Required. The name of the object type that receives data from the source fields. Valid values are: • `WorkPlan` • `WorkStep`

ObjectMappingField

A field name in the input object and the corresponding field name in the output object.

Field Name	Field Type	Description
`inputField`	string	Required. The name of a custom field supplying source data. This field is from the object specified in `inputObject`.
`outputField`	string	Required. The name of a custom field that receives data from the source field specified in `inputField`. This field is from the object specified in `outputObject`.

Declarative Metadata Sample Definition

This sample file shows a subset of the possible field service settings that you can customize.

```
<?xml version="1.0" encoding="UTF-8"?>
<FieldServiceSettings xmlns="http://soap.sforce.com/2006/04/metadata">
   <doesAllowEditSaForCrew>false</doesAllowEditSaForCrew>
   <doesShareSaParentWoWithAr>false</doesShareSaParentWoWithAr>
   <doesShareSaWithAr>false</doesShareSaWithAr>
   <enableWorkOrders>false</enableWorkOrders>
   <enableWorkPlansAutoGeneration>true</enableWorkPlansAutoGeneration>
   <fieldServiceNotificationsOrgPref>false</fieldServiceNotificationsOrgPref>
   <fieldServiceOrgPref>true</fieldServiceOrgPref>
   <isGeoCodeSyncEnabled>false</isGeoCodeSyncEnabled>
   <isLocationHistoryEnabled>false</isLocationHistoryEnabled>
   <o2EngineEnabled>false</o2EngineEnabled>
   <objectMappingItem>
      <mappingType>WorkPlans_WorkPlanTemplate_WorkPlan</mappingType>
```

```
      <objectMapping>
         <inputObject>WorkPlanTemplate</inputObject>
         <mappingFields>
            <inputField>WorkPlanTemplate_CustomNumberField__c</inputField>
            <outputField>WorkPlan_CustomNumberField__c</outputField>
         </mappingFields>
         <mappingFields>
            <inputField>WorkPlanTemplate_CustomTextField__c</inputField>
            <outputField>WorkPlan_CustomPicklistField__c</outputField>
         </mappingFields>
         <outputObject>WorkPlan</outputObject>
      </objectMapping>
   </objectMappingItem>
   <objectMappingItem>
      <mappingType>WorkPlans_WorkStepTemplate_WorkStep</mappingType>
      <objectMapping>
         <inputObject>WorkStepTemplate</inputObject>
         <mappingFields>
            <inputField>WokStepTemplate_CustomNumberField__c</inputField>
            <outputField>WokStep_CustomNumberField__c</outputField>
         </mappingFields>
         <mappingFields>
            <inputField>WokStepTemplate_CustomTextField__c</inputField>
            <outputField>WokStep_CustomTextField__c</outputField>
         </mappingFields>
         <outputObject>WorkStep</outputObject>
      </objectMapping>
   </objectMappingItem>
   <objectMappingItem>
      <mappingType>WorkPlans_WorkPlanTemplateEntry_WorkStep</mappingType>
      <objectMapping>
         <inputObject>WorkPlanTemplateEntry</inputObject>
         <mappingFields>
            <inputField>WorkPlanTemplateEntry_CustomDateField__c</inputField>
            <outputField>WokStep_CustomDateField__c</outputField>
         </mappingFields>
         <outputObject>WorkStep</outputObject>
      </objectMapping>
   </objectMappingItem>
   <optimizationServiceAccess>false</optimizationServiceAccess>
   <serviceAppointmentsDueDateOffsetOrgValue>7</serviceAppointmentsDueDateOffsetOrgValue>

   <workOrderLineItemSearchFields>Subject</workOrderLineItemSearchFields>
   <workOrderSearchFields>Subject</workOrderSearchFields>
</FieldServiceSettings>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

FilesConnectSettings

Represents the settings that modify the Files Connect feature.This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

`FilesConnectSettings` values are stored in the `FilesConnect.settings` file in the `settings` folder. The `.settings` files are different from other named components, because there's only one settings file for each settings component.

Version

FilesConnectSettings components are available in API version 47.0 and later.

Fields

Field Name	Field Type	Description
`enableContentHubAllowed`	boolean	Indicates whether Files Connect is enabled (`true`) or not (`false`).
`enableContentHubCvtLinksAllowed`	boolean	Indicates whether the ability to link conversions in the feed publisher is enabled (`true`) or not (`false`) for Google Drive and Quip links.
`enableContentHubEOSearchLayout`	boolean	Indicates whether the external object's search layout can be used in Global Search (`true`) or not (`false`).

Declarative Metadata Sample Definition

The following is an example of a FilesConnectSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<FilesConnectSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <enableContentHubAllowed>false</enableCurrencyEffectiveDates>
    <enableContentHubCvtLinksAllowed>false</enableCurrencySymbolWithMultiCurrency>
    <enableContentHubEOSearchLayout>false</enableMultiCurrency>
</FilesConnectSettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>FilesConnect</members>
        <name>Settings</name>
    </types>
    <version>47.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

FileUploadAndDownloadSecuritySettings

Represents the security settings for uploading and downloading files. This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

FileUploadAndDownloadSecuritySettings components have the suffix `.settings` and are stored in the `settings` folder.

Version

FileUploadAndDownloadSecuritySettings components are available in API version 39.0 and later.

Fields

Field Name	Field Type	Description
`dispositions`	FileTypeDispositionAssignmentBean[]	Represents the metadata used to manage filetype behavior. This field is available in API version 39.0 and later.
`noHtmlUploadAsAttachment`	boolean	Indicates whether to allow HTML uploads as attachments or document records. This field is available in API version 39.0 and later.

FileTypeDispositionAssignmentBean

Represents the metadata used to manage filetype behavior.

Field Name	Field Type	Description
`behavior`	FileDownloadBehavior (enumeration of type string)	One of the following values: • DOWNLOAD • EXECUTE • HYBRID The following filetypes are a security risk and can not have EXECUTE behavior: • EXE • FLASH • HTML • RFC822

Field Name	Field Type	Description
		• SVG • TXML • UNKNOWN • WEBVIEW • XHTML • XML
`filetype`	FileType (enumeration of type string)	Although more filetypes exist, these are the only ones supported by FileTypeDispositionAssignmentBean: • AVI • EXCEL • EXCEL_X • EXE • FLASH • HTML • INSIGHT • MOV • MP3 • MP4 • MPEG • PDF • POWER_POINT • POWER_POINT_X • RFC822 • SVG • TXML • UNKNOWN • WAV • WEBVIEW • WMA • WMV • WORD • WORD_X • XHTML • XML
`securityRiskFileType`	boolean	Indicates filetypes that cannot have behavior set to EXECUTE, due to security risks. This field is read-only.

Declarative Metadata Sample Definition

The following is an example of a FileUploadAndDownloadSecuritySettings component.

```
<FileUploadAndDownloadSecuritySettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <dispositions>
        <behavior>HYBRID</behavior>
        <fileType>AVI</fileType>
        <securityRiskFileType>false</securityRiskFileType>
    </dispositions>
    <dispositions>
        <behavior>HYBRID</behavior>
        <fileType>WORD</fileType>
        <securityRiskFileType>false</securityRiskFileType>
    </dispositions>
    <dispositions>
        <behavior>HYBRID</behavior>
        <fileType>WORD_X</fileType>
        <securityRiskFileType>false</securityRiskFileType>
    </dispositions>
    <dispositions>
        <behavior>DOWNLOAD</behavior>
        <fileType>EXE</fileType>
        <securityRiskFileType>true</securityRiskFileType>
    </dispositions>
    <dispositions>
        <behavior>DOWNLOAD</behavior>
        <fileType>HTML</fileType>
        <securityRiskFileType>true</securityRiskFileType>
    </dispositions>
    <dispositions>
        <behavior>DOWNLOAD</behavior>
        <fileType>WEBVIEW</fileType>
        <securityRiskFileType>true</securityRiskFileType>
    </dispositions>
    <dispositions>
        <behavior>DOWNLOAD</behavior>
        <fileType>RFC822</fileType>
        <securityRiskFileType>true</securityRiskFileType>
    </dispositions>
    <dispositions>
        <behavior>HYBRID</behavior>
        <fileType>MOV</fileType>
        <securityRiskFileType>false</securityRiskFileType>
    </dispositions>
    <dispositions>
        <behavior>HYBRID</behavior>
        <fileType>MP3</fileType>
        <securityRiskFileType>false</securityRiskFileType>
    </dispositions>
    <dispositions>
        <behavior>HYBRID</behavior>
        <fileType>MP4</fileType>
        <securityRiskFileType>false</securityRiskFileType>
    </dispositions>
```

```
    <dispositions>
        <behavior>HYBRID</behavior>
        <fileType>MPEG</fileType>
        <securityRiskFileType>false</securityRiskFileType>
    </dispositions>
    <dispositions>
        <behavior>HYBRID</behavior>
        <fileType>PDF</fileType>
        <securityRiskFileType>false</securityRiskFileType>
    </dispositions>
    <dispositions>
        <behavior>HYBRID</behavior>
        <fileType>POWER_POINT</fileType>
        <securityRiskFileType>false</securityRiskFileType>
    </dispositions>
    <dispositions>
        <behavior>HYBRID</behavior>
        <fileType>POWER_POINT_X</fileType>
        <securityRiskFileType>false</securityRiskFileType>
    </dispositions>
    <dispositions>
        <behavior>DOWNLOAD</behavior>
        <fileType>SVG</fileType>
        <securityRiskFileType>true</securityRiskFileType>
    </dispositions>
    <dispositions>
        <behavior>DOWNLOAD</behavior>
        <fileType>FLASH</fileType>
        <securityRiskFileType>true</securityRiskFileType>
    </dispositions>
    <dispositions>
        <behavior>DOWNLOAD</behavior>
        <fileType>TXML</fileType>
        <securityRiskFileType>true</securityRiskFileType>
    </dispositions>
    <dispositions>
        <behavior>DOWNLOAD</behavior>
        <fileType>UNKNOWN</fileType>
        <securityRiskFileType>true</securityRiskFileType>
    </dispositions>
    <dispositions>
        <behavior>HYBRID</behavior>
        <fileType>WAV</fileType>
        <securityRiskFileType>false</securityRiskFileType>
    </dispositions>
    <dispositions>
        <behavior>HYBRID</behavior>
        <fileType>WMA</fileType>
        <securityRiskFileType>false</securityRiskFileType>
    </dispositions>
    <dispositions>
        <behavior>HYBRID</behavior>
        <fileType>WMV</fileType>
        <securityRiskFileType>false</securityRiskFileType>
```

```
    </dispositions>
    <dispositions>
        <behavior>DOWNLOAD</behavior>
        <fileType>XHTML</fileType>
        <securityRiskFileType>true</securityRiskFileType>
    </dispositions>
    <dispositions>
        <behavior>HYBRID</behavior>
        <fileType>EXCEL</fileType>
        <securityRiskFileType>false</securityRiskFileType>
    </dispositions>
    <dispositions>
        <behavior>HYBRID</behavior>
        <fileType>EXCEL_X</fileType>
        <securityRiskFileType>false</securityRiskFileType>
    </dispositions>
    <dispositions>
        <behavior>DOWNLOAD</behavior>
        <fileType>XML</fileType>
        <securityRiskFileType>true</securityRiskFileType>
    </dispositions>
    <noHtmlUploadAsAttachment>false</noHtmlUploadAsAttachment>
</FileUploadAndDownloadSecuritySettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
   <types>
       <members>FileUploadAndDownloadSecurity</members>
       <name>Settings</name>
   </types>
   <version>39.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

FlowSettings

Represents the Salesforce settings for processes and flows, such as whether Lightning runtime for flows is enabled.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

FlowSettings values are stored in the `Flow.settings` file in the settings directory. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

FlowSettings components are available in API version 47.0 and later.

Fields

Field Name	Field Type	Description
`canDebugFlowAsAnotherUser`	boolean	Indicates whether a flow can be debugged as another user (`true`) or not (`false`). Corresponds to the `Let admins debug flows as other users` field on the Process Automation Settings page in Setup. Available in API version 50.0 and later.
`doesEnforceApexCpuTimeLimit`	boolean	Indicates whether Salesforce accurately measures the CPU time that flows and processes consume (`true`) or not (`false`). Corresponds to the Accurately Measure the CPU Time Consumption of Flows and Processes release update. Available in API version 51.0 and later.
`doesFormulaEnforceDataAccess`	boolean	Indicates whether formula resources and formula fields in a flow enforce record-level security (`true`) or not (`false`). Corresponds to the Enforce Data Access in Flow Formulas critical update. Available in API version 48.0 and later.
`doesFormulaGenerateHtmlOutput`	boolean	Indicates whether flow formula functions that generate HTML, such as `BR()`, `IMAGE()`, and `HYPERLINK()`, include encoded markers (`__BR_ENCODED__`) (`true`) or not (`false`). Available in API version 48.0 and later.
`enableFlowBREncodedFixEnabled`	boolean	Indicates whether `BR()` functions in flow and process formulas result in a line break (`true`) or resolve to `_BR_ENCODED_` as a literal value (`false`). Corresponds to the Use the BR() Function in Flows and Processes critical update.
`enableFlowCustomPropertyEditor`	boolean	Indicates whether an org has custom property editors enabled for actions and screen fields in Flow Builder (`true`) or not (`false`). This field is available in API version 48.0 to 50.0. The field is deprecated in API version 50.0 and later. All orgs that have Flow Builder support custom property editors for actions and screen fields.
`enableFlowDeployAsActiveEnabled`	boolean	Indicates whether processes and flows can be deployed as active via change sets or Metadata API. When the value is `false`, all processes and flows are deployed as inactive. When the value is `true`, deploying an active process or flow in a production org causes your Apex tests to run. If Apex tests don't launch your org's required percentage of active processes and autolaunched flows, the deployment is rolled back.

Field Name	Field Type	Description
		The default value is `false` for production orgs and is `true` for non-production orgs such as scratch, sandbox, and developer orgs. Corresponds to the `Deploy processes and flows as active` field on the Process Automation Settings page in Setup. The field appears in the user interface on production orgs only.
`enableFlowFieldFilterEnabled`	boolean	Indicates whether flows can successfully execute Create Records and Update Records elements that update fields to which the running doesn't have edit access. By default (`false`), the Create Records or Update Records element fails and executes the fault path if it has one. When the value is `true`, the element sets only the fields that the running user can edit. No notification is sent when some fields aren't updated. Corresponds to the `Filter inaccessible fields from flow requests` field on the Process Automation Settings page in Setup.
`enableFlowFormulasFixEnabled`	boolean	Indicates whether process and flow formulas return null values when the calculations involve a null record variable or null lookup relationship field. When the value is `true`, those formulas return null values at run time. When the value is `false`, those formulas return unhandled exceptions at run time. Corresponds to the Check for Null Record Variables or Null Values of Lookup Relationship Fields in Process and Flow Formulas critical update.
`enableFlowInterviewSharingEnabled`	boolean	Indicates whether users can resume the paused flow interviews that they have edit access to. By default (`true`), users can resume interviews that are shared with them, either directly or via the role hierarchy. When the value is `false`, each paused interview can be resumed only by the interview owner or a flow admin who has view access to the interview. Corresponds to the `Let users resume shared flow interviews` field on the Process Automation Settings page in Setup.
`enableFlowNullPreviousValueFix`	boolean	Indicates whether each process evaluates criteria by always using the original record field values from when the process begins. When the value is `true`, each process with an Update Records action and multiple criteria nodes always evaluates criteria using the original field values of the record. When the value is `false`, processes evaluate the updated values of record fields that were null when the process began. Corresponds to the Evaluate Criteria Based on Original Record Values in the Process Builder critical update.
`enableFlowPauseEnabled`	boolean	Indicates whether screens can display the Pause button so that users can pause flow interviews. By default, the value is `false`. Corresponds to the `Let users pause flows` field on the Process Automation Settings page in Setup.

Field Name	Field Type	Description
`enableFlowReactiveScreens`	boolean	Indicates whether supported screen components in flows running on API version 57.0 and 58.0 can react to changes in other components on the same screen. This setting isn't applicable to flows running on API version 59.0 and later. By default, the value is `false`. To make a component reactive, reference the output of another component on the same screen in the configuration pane. Corresponds to the `Enable Reactive Components for Specific Flow Versions` field on the Process Automation Settings page in Setup.
`enableFlowUseApexExceptionEmail`	boolean	Indicates whether process and flow error emails are sent to: • The user who last modified the process or flow (`false`) • The addresses set on the Apex Exception Email page in Setup (`true`) By default, the value is `false`. Corresponds to the `Send Process or Flow Error Email to` field on the Process Automation Settings page in Setup.
`enableFlowViaRestUsesUserCtxt`	boolean	Indicates whether a flow that runs via REST API uses the running user's profile and permission sets to determine the object permissions and field-level access of the flow. Corresponds to the Run Flows in User Context via REST API critical update. Available in API version 54.0 and later.
`enableInvocableFlowFixEnabled`	boolean	Removed in API version 50.0 and later. Indicates whether all autolaunched flow interviews are executed when they're invoked in bulk from a process or the Invocable Actions resource in REST API (`true`) or not (`false`). When the value is `false`, flow interviews that share identical input parameters aren't executed. Corresponds to the Execute All Flow Interviews When Invoked in Bulk critical update.
`enableLightningRuntimeEnabled`	boolean	Indicates whether flows that are launched from a URL or from Setup use the Lightning runtime experience (`true`) or the Classic runtime experience (`false`). By default, the value is `true`. Corresponds to the `Enable Lightning runtime for flows` field on the Process Automation Settings page in Setup.
`isAccessToInvokedApexRequired`	boolean	Indicates whether flows can invoke Apex classes only when the running users' profiles or permission sets include access to those Apex classes. When the value is `false`, Apex class security doesn't apply to flows. Corresponds to the Require User Access to Apex Classes Invoked by Flow critical update.

Field Name	Field Type	Description
		This field is available in API versions 47.0 to 58.0. The field is deprecated in API version 59.0 and later.
`isApexPluginAccessModifierRespected`	boolean	Indicates whether flows respect the public access modifiers for legacy Apex actions. When the value is `true`: • Flows fail when they execute public legacy Apex actions from a different namespace. • Public legacy Apex actions from a different namespace aren't available in Flow Builder. • Global legacy Apex actions with public `describe` or `invoke` methods are unavailable to flows in a different namespace. When the value is `false`, you can add public legacy Apex actions to flows even though they're not supported. Also, global legacy Apex actions with public `describe` or `invoke` methods are available to flows in a different namespace. Corresponds to the Make Flows Respect Access Modifiers for Legacy Apex Actions critical update. Available in API version 48.0 and later.
`isEnhancedFlowListViewVisible`	boolean	Indicates whether the enhanced Flows list view in Lightning Experience replaces the Classic Flows list view (`true`) or not (`false`). The default value is `true`. If the field is set to `false`, the Classic Flows list view replaces the enhanced list view.
`isFlowApexContextRetired`	boolean	Indicates whether the rules for enforcing explicit access to Apex classes are disabled (`true`) or not (`false`). For most Salesforce orgs the default value is `true`. Corresponds to the Disable Rules for Enforcing Explicit Access to Apex Classes release update. This field is available in API versions 49.0 to 58.0. The field is deprecated in API version 59.0 and later.
`isFlowBlockAccessToSessionIDEnabled`	boolean	Indicates whether a valid session ID is returned in API.SessionID (`false`) or not (`true`). The default value is false. When the value is `true`, flows that access the session ID variable receive a placeholder string instead of a valid session ID.
`isManageFlowRequiredForAutomationCharts`	boolean	Indicates whether the Manage Flow permission is required to view all charts in Automation Home (Beta) (`true`) or not (`false`). The default value is `false`. All users with the View Setup and Configuration permission can view all charts in Automation Home. If the field is set to `true`, then users with the View Setup and Configuration permission can view only the Total Started Automations by Process Type chart. The Manage Flow permission is required to view all charts.

Field Name	Field Type	Description
`isTimeResumedInSameRunContext`	boolean	Indicates whether paused autolaunched flows always resume in the same context and retain the user access that they had before being paused (`true`) or not (`false`). Corresponds to the Make Paused Flow Interviews Resume in the Same Context with the Same User Access release update. This field is available in API version 57.0 and later.

Declarative Metadata Sample Definition

Here's an example of the Flow.settings file.

```
<?xml version="1.0" encoding="UTF-8"?>
<FlowSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <doesFormulaEnforceDataAccess>true</doesFormulaEnforceDataAccess>
    <enableFlowBREncodedFixEnabled>true</enableFlowBREncodedFixEnabled>
    <enableFlowDeployAsActiveEnabled>false</enableFlowDeployAsActiveEnabled>
    <enableFlowFieldFilterEnabled>false</enableFlowFieldFilterEnabled>
    <enableFlowFormulasFixEnabled>true</enableFlowFormulasFixEnabled>
    <enableFlowInterviewSharingEnabled>true</enableFlowInterviewSharingEnabled>
    <enableFlowNullPreviousValueFix>true</enableFlowNullPreviousValueFix>
    <enableFlowPauseEnabled>true</enableFlowPauseEnabled>
    <enableFlowUseApexExceptionEmail>false</enableFlowUseApexExceptionEmail>
    <enableLightningRuntimeEnabled>true</enableLightningRuntimeEnabled>
    <isApexPluginAccessModifierRespected>true</isApexPluginAccessModifierRespected>
    <isEnhancedFlowListViewVisible>true</isEnhancedFlowListViewVisible>
   <isManageFlowRequiredForAutomationCharts>false</isManageFlowRequiredForAutomationCharts>
</FlowSettings>
```

Example Package Manifest

Here's an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
 <types>
  <members>Flow</members>
  <name>Settings</name>
 </types>
 <version>47.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ForecastingObjectListSettings

Represents an org's forecasting object list settings. Use these settings to control which object types and field types appear in the list of object details on the forecasts page. For example, pipeline forecasts use the Opportunity object, and the object list settings specify which fields from that object are available in the opportunity list section of the forecasts page. This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

ForecastingObjectListSettings values are stored in the `ForecastingObjectList.settings` file in the `settings` folder. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

ForecastingObjectListSettings is available in API versions 52.0 and later.

Note: The information in this topic applies only to forecast types created in Summer '21 and later.

Fields

Field Name	Field Type	Description
`forecastingTypeObjectListSettings`	ForecastingTypeObjectListSettings	For each forecast type, specifies the information that is displayed in the list of object details that roll up into the forecasts. For example, the list of opportunities rolls up into opportunity-based forecasts.

ForecastingTypeObjectListSettings

Represents an org's forecasting type object list settings.

Field Name	Field Type	Description
`forecastingObjectListLabelMappings`	ForecastingObjectListLabelMapping	Mapping of labels with each field displayed as a column in the object detail list on the forecasts page.
`forecastingObjectListSelectedSettings`	forecastingObjectListSelectedSetting	Specifies the object fields that are used as columns in the object detail list on the forecasts page.
`forecastingObjectListUnselectedSettings`	forecastingObjectListUnselectedSetting	Lists the object fields that are available but not currently used as columns in the object detail list on the forecasts page. Changes to `forecastingObjectListSelectedSettings` field are reflected in this field.
`forecastingTypeDeveloperName`	string	Developer name of the forecast type that these object list settings apply to.

ForecastingObjectListLabelMapping

Represents an org's forecasting type object list label mapping.

Field Name	Field Type	Description
field	string	Object field's API name.
label	string	Object field's name in the object detail list on the forecasts page.

forecastingObjectListSelectedSettings

Represents an org's forecasting type object list selected settings.

Field Name	Field Type	Description
field	string	Object field's API name.

forecastingObjectListUnselectedSettings

Represents an org's forecasting type object list unselected settings.

Field Name	Field Type	Description
field	string	Object field's API name.

Declarative Metadata Sample Definition

The following is an example of the ForecastingObjectListSettings.settings file:

```
<?xml version="1.0" encoding="UTF-8"?>
<ForecastingObjectListSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <forecastingTypeObjectListSettings>
        <forecastingObjectListLabelMappings>
            <field>CREATEDBY_USER.ALIAS</field>
            <label>Created By Alias</label>
        </forecastingObjectListLabelMappings>
        <forecastingObjectListLabelMappings>
            <field>OPPORTUNITY.AMOUNT</field>
            <label>Amount</label>
        </forecastingObjectListLabelMappings>
        <forecastingObjectListLabelMappings>
            <field>OPPORTUNITY.CLOSE_DATE</field>
            <label>Close Date</label>
        </forecastingObjectListLabelMappings>
        <forecastingObjectListLabelMappings>
            <field>OPPORTUNITY.TYPE</field>
            <label>Type</label>
        </forecastingObjectListLabelMappings>
        <forecastingObjectListLabelMappings>
            <field>OPPORTUNITY.CREATED_DATE</field>
```

```
        <label>Created Date</label>
    </forecastingObjectListLabelMappings>
    <forecastingObjectListLabelMappings>
        <field>OPPORTUNITY.LAST_UPDATE</field>
        <label>Last Modified Date</label>
    </forecastingObjectListLabelMappings>
    <forecastingObjectListLabelMappings>
        <field>OPPORTUNITY.LEAD_SOURCE</field>
        <label>Lead Source</label>
    </forecastingObjectListLabelMappings>
    <forecastingObjectListLabelMappings>
        <field>OPPORTUNITY.EXP_AMOUNT</field>
        <label>Expected Revenue</label>
    </forecastingObjectListLabelMappings>
    <forecastingObjectListLabelMappings>
        <field>OPPORTUNITY.CLOSED</field>
        <label>Closed</label>
    </forecastingObjectListLabelMappings>
    <forecastingObjectListLabelMappings>
        <field>OPPORTUNITY.WON</field>
        <label>Won</label>
    </forecastingObjectListLabelMappings>
    <forecastingObjectListLabelMappings>
        <field>00Nxx000001G2W0</field>
        <label>CustomOppCurr</label>
    </forecastingObjectListLabelMappings>
    <forecastingObjectListLabelMappings>
        <field>CORE.USERS.ALIAS</field>
        <label>Opportunity Owner Alias</label>
    </forecastingObjectListLabelMappings>
    <forecastingObjectListLabelMappings>
        <field>OPPORTUNITY.PROBABILITY</field>
        <label>Probability (%)</label>
    </forecastingObjectListLabelMappings>
    <forecastingObjectListLabelMappings>
        <field>OPPORTUNITY.LAST_ACTIVITY</field>
        <label>Last Activity</label>
    </forecastingObjectListLabelMappings>
    <forecastingObjectListLabelMappings>
        <field>OPPORTUNITY.FISCAL_QUARTER</field>
        <label>Fiscal Quarter</label>
    </forecastingObjectListLabelMappings>
    <forecastingObjectListLabelMappings>
        <field>00Nxx000001G8GS</field>
        <label>TaraTestOppCurr</label>
    </forecastingObjectListLabelMappings>
    <forecastingObjectListLabelMappings>
        <field>DESCRIPTION</field>
        <label>Description</label>
    </forecastingObjectListLabelMappings>
    <forecastingObjectListLabelMappings>
        <field>OPPORTUNITY.FISCAL_PERIOD</field>
        <label>Fiscal Period</label>
    </forecastingObjectListLabelMappings>
```

```
    <forecastingObjectListLabelMappings>
        <field>FULL_NAME</field>
        <label>Owner Full Name</label>
    </forecastingObjectListLabelMappings>
    <forecastingObjectListLabelMappings>
        <field>OPPORTUNITY.NEXT_STEP</field>
        <label>Next Step</label>
    </forecastingObjectListLabelMappings>
    <forecastingObjectListLabelMappings>
        <field>UPDATEDBY_USER.ALIAS</field>
        <label>Last Modified By Alias</label>
    </forecastingObjectListLabelMappings>
    <forecastingObjectListLabelMappings>
        <field>OPPORTUNITY.STAGE_NAME</field>
        <label>Stage</label>
    </forecastingObjectListLabelMappings>
    <forecastingObjectListLabelMappings>
        <field>CONTRACT.NAME</field>
        <label>Contract Name</label>
    </forecastingObjectListLabelMappings>
    <forecastingObjectListLabelMappings>
        <field>OPPORTUNITY.QUANTITY</field>
        <label>Quantity</label>
    </forecastingObjectListLabelMappings>
    <forecastingObjectListLabelMappings>
        <field>SPLITAMOUNT</field>
        <label>Forecasted Amount</label>
    </forecastingObjectListLabelMappings>
    <forecastingObjectListLabelMappings>
        <field>OPPORTUNITY.NAME</field>
        <label>Opportunity Name</label>
    </forecastingObjectListLabelMappings>
    <forecastingObjectListLabelMappings>
        <field>CORE.USERS.LAST_NAME</field>
        <label>Owner Last Name</label>
    </forecastingObjectListLabelMappings>
    <forecastingObjectListLabelMappings>
        <field>OPPORTUNITY.FISCAL_YEAR</field>
        <label>Fiscal Year</label>
    </forecastingObjectListLabelMappings>
    <forecastingObjectListLabelMappings>
        <field>TERR2_NAME</field>
        <label>Territory Name</label>
    </forecastingObjectListLabelMappings>
    <forecastingObjectListLabelMappings>
        <field>CORE.USERS.FIRST_NAME</field>
        <label>Owner First Name</label>
    </forecastingObjectListLabelMappings>
    <forecastingObjectListLabelMappings>
        <field>ACCOUNT.SITE</field>
        <label>Account Site</label>
    </forecastingObjectListLabelMappings>
    <forecastingObjectListLabelMappings>
        <field>ACCOUNT.NAME</field>
```

```
        <label>Account Name</label>
    </forecastingObjectListLabelMappings>
    <forecastingObjectListLabelMappings>
        <field>OPPORTUNITY.PRIVATE</field>
        <label>Private</label>
    </forecastingObjectListLabelMappings>
    <forecastingObjectListLabelMappings>
        <field>TERR2_DESC</field>
        <label>Territory Description</label>
    </forecastingObjectListLabelMappings>
    <forecastingObjectListLabelMappings>
        <field>CONTRACT.CONTRACT_NUMBER</field>
        <label>Contract Number</label>
    </forecastingObjectListLabelMappings>
    <forecastingObjectListLabelMappings>
        <field>FORECAST_CATEGORY</field>
        <label>Forecast Category</label>
    </forecastingObjectListLabelMappings>
    <forecastingObjectListSelectedSettings>
        <field>OPPORTUNITY.NAME</field>
    </forecastingObjectListSelectedSettings>
    <forecastingObjectListUnselectedSettings>
        <field>ACCOUNT.NAME</field>
        <field>CONTRACT.CONTRACT_NUMBER</field>
        <field>CONTRACT.NAME</field>
        <field>OPPORTUNITY.STAGE_NAME</field>
        <field>FORECAST_CATEGORY</field>
        <field>OPPORTUNITY.CLOSE_DATE</field>
        <field>OPPORTUNITY.AMOUNT</field>
        <field>CORE.USERS.ALIAS</field>
        <field>CORE.USERS.FIRST_NAME</field>
        <field>CORE.USERS.LAST_NAME</field>
        <field>FULL_NAME</field>
        <field>OPPORTUNITY.PROBABILITY</field>
        <field>DESCRIPTION</field>
        <field>OPPORTUNITY.EXP_AMOUNT</field>
        <field>OPPORTUNITY.LEAD_SOURCE</field>
        <field>OPPORTUNITY.NEXT_STEP</field>
        <field>OPPORTUNITY.PRIVATE</field>
        <field>OPPORTUNITY.QUANTITY</field>
        <field>OPPORTUNITY.TYPE</field>
        <field>UPDATEDBY_USER.ALIAS</field>
        <field>CREATEDBY_USER.ALIAS</field>
        <field>OPPORTUNITY.CLOSED</field>
        <field>OPPORTUNITY.WON</field>
        <field>ACCOUNT.SITE</field>
        <field>OPPORTUNITY.FISCAL_YEAR</field>
        <field>OPPORTUNITY.FISCAL_QUARTER</field>
        <field>OPPORTUNITY.FISCAL_PERIOD</field>
        <field>OPPORTUNITY.LAST_ACTIVITY</field>
        <field>OPPORTUNITY.CREATED_DATE</field>
        <field>OPPORTUNITY.LAST_UPDATE</field>
        <field>SPLITAMOUNT</field>
        <field>00Nxx000001G2W0</field>
```

```
            <field>00Nxx000001G8GS</field>
            <field>TERR2_NAME</field>
            <field>TERR2_DESC</field>
        </forecastingObjectListUnselectedSettings>

<forecastingTypeDeveloperName>OpportunityLineItemRevenue</forecastingTypeDeveloperName>
    </forecastingTypeObjectListSettings>
```

Example Package Manifest

The following is an example package manifest used to deploy or retrieve the ForecastingObjectListSettingsSettings settings metadata:

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
 <types>
  <members>ForecastingObjectListSettings</members>
  <name>Settings</name>
 </types>
 <version>52.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ForecastingSettings

Represents the Forecasts settings options. This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Note: ForecastingSettings components are available in API version 28 and later. The structure of the ForecastingSettings type changed significantly in API version 30.0 and in API version 53.0.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

ForecastingSettings values are stored in a single file named `Forecasting.settings` in the `settings` directory of the corresponding package directory. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

ForecastingSettings components are available in API version 28 and later. The structure of the ForecastingSettings type changed significantly in API version 30.0 and in API version 53.0.

Fields

Field Name	Field Type	Description
`displayCurrency`	DisplayCurrency (enumeration of type string)	Removed. The currency for displaying forecasts; either the organization's corporate currency or each forecast owner's personal currency setting. The selected currency is the default used in forecasts and selected in setup. The selection must be one of the currencies enabled for use in the organization, and only one selection is allowed. The default is `CORPORATE`. The valid values are: • `CORPORATE` • `PERSONAL` Available in API version 28.0 to 46.0. In API version 47.0 and later, use `defaultToPersonalCurrency`.
`defaultToPersonalCurrency`	boolean	If multicurrency is enabled, this field indicates whether the user's personal currency is used in forecasts. If `true` (default), the user's personal currency is used. If `false`, the corporate currency is used. Available in API version 47.0 and later.
`enableForecasts`	boolean	Indicates if Salesforce Forecasting is enabled or not. Set to `true` to enable and `false` to disable the functionality. Disabling Forecasts can result in data loss. Refer to Salesforce Help before disabling any functionality.
`forecastingCategoryMappings`	ForecastingCategoryMapping[]	A list of mappings associating forecast types with forecast rollups. As of Spring '20 and later, only standard users with the View All Forecasts or Allow Forecasting permission or delegated forecast manager status can access this subtype.
`forecastingDisplayedFamilySettings`	ForecastingDisplayedFamilySettings[]	The product families chosen to allow forecasting on in Lightning Experience. This field is available in API version 40.0 and later.
`forecastingSubmissionSettings`	ForecastingSubmissionSettings	The submission options for forecasts. Available in API version 62.0 and later.
`forecastingTypeSettings`	ForecastingTypeSettings[]	A list of forecast types. For field values, see ForecastingTypeSettings. The maximum number of forecast types is four.
`globalAdjustmentsSettings`	AdjustmentsSettings[]	The adjustment options for forecasts. Available in API version 53.0 and later. In API version 52.0 and earlier, use the `adjustmentsSettings` field on ForecastingTypeSettings.
`globalForecastRangeSettings`	ForecastRangeSettings[]	The default periods and range selections in forecasts. Available in API version 53.0 and later. In API version 52.0 and earlier, use the `forecastRangeSettings` field on ForecastingTypeSettings.
`globalQuotasSettings`	QuotasSettings[]	Enables or disables quotas in Salesforce Forecasting. Available in API version 53.0 and later. In API version 52.0 and earlier, use the `quotasSettings` field on ForecastingTypeSettings.

ForecastingCategoryMapping

The category mappings for forecasts. This subtype appears eight times within the `ForecastingSettings` type. Each occurrence includes fields that specify a type of forecast category rollup, which forecast categories each rollup includes, and the weight of each forecast category in the rollup. Organizations using either cumulative forecast rollups or individual forecast category columns must include all eight occurrences of this subtype. As of Spring '20 and later, only standard users with the View All Forecasts or Allow Forecasting permission or delegated forecast manager status can access this subtype.

Field	Field Type	Description
`forecastingItemCategoryApiName`	string	Required. This field specifies the API name of the rollup type. The valid values are: • `openpipeline` • `bestcaseforecast` • `commitforecast` • `pipelineonly` • `bestcaseonly` • `commitonly` • `closedonly` • `omittedonly` • `customcategory`
`weightedSourceCategories`	WeightedSourceCategory[]	This field can occur more than one time when specifying more than one forecast category to include in the rollup type. Each occurrence contains two subfields that specify a forecast category to include in the forecast rollup type and its weight. Some rollup types include more than one forecast category. This list shows the forecast categories that are included in each rollup type. • Rollup: `openpipeline`, Forecast categories: pipeline, best case, commit • Rollup: `bestcaseforecast`, Forecast categories: best case, commit, closed • Rollup: `commitforecast`, Forecast categories: commit, closed • Rollup: `pipelineonly`, Forecast categories: `pipeline` • Rollup: `bestcaseonly`, Forecast categories: `best case` • Rollup: `commitonly`, Forecast categories: `commit` • Rollup: `closedonly`, Forecast categories: `closed` • Rollup: `omittedonly`, Forecast categories: `omitted` • Rollup: customcategory, Forecast categories: custom category

ForecastingDisplayedFamilySettings

The product families that an admin chooses to allow forecasting on in Lightning Experience. This field is available in API version 40.0 and later.

Field	Field Type	Description
`productFamily`	string	The product family available to forecast on. Each product family is unique.

ForecastingSubmissionSettings

The setting allows forecast users to submit their forecast numbers at a point in time.

Field	Field Type	Description
`allowForecastingSubmissions`	boolean	Required. Indicates whether forecast submissions are allowed in forecasts.

ForecastingTypeSettings

The settings for each forecast type. An organization can have up to four forecast types active. If you omit a previously enabled forecast type that has a minimum API version less than or equal to the metadata package version, its quota and adjustment data is deleted from the org.

Omitting a forecast type field from the XML can deactivate that forecast type: if the forecast type was available in the release specified by the XML package version, that forecast type is deactivated and its quota and adjustment data are deleted.

Field Name	Field Type	Description
`active`	boolean	Required. Indicates whether the forecast type specified in the `name` field is active. Setting the `active` field to false purges all forecasting data, adjustments, and quotas for the forecast type. When `active` is set to true, some values on the Forecasts tab don't appear immediately. An in-process con appears to indicate that the values are being calculated.
`adjustmentsSettings`	AdjustmentsSettings	Removed. This field enables or disables the adjustments option in forecasts. In API version 53.0 and later, use globalAdjustmentsSettings.
`displayedCategoryApiNames`	string	This read-only field appears four times to specify the four forecast rollup categories displayed in the Forecasts tab, for either cumulative forecast rollups, or individual forecast category rollups. Always use the same 4 values for both `displayedCategoryApiNames` and `forecastedCategoryApiNames`. Valid values for organizations using cumulative forecast rollups: • `openpipeline` • `bestcaseforecast` • `commitforecast`

Field Name	Field Type	Description
		• `closedonly` Valid values for organizations using individual forecast category rollups: • `pipelineonly` • `bestcaseonly` • `commitonly` • `closedonly`
`forecastedCategoryApiNames`	string	This field appears four times to specify the four forecast rollup categories used in the organization, for either cumulative forecast rollups, or individual forecast category rollups. Valid values for organizations using cumulative forecast rollups: • `openpipeline` • `bestcaseforecast` • `commitforecast` • `closedonly` • `customcategory` Valid values for organizations using individual forecast category rollups: • `pipelineonly` • `bestcaseonly` • `commitonly` • `closedonly` • `customcategory` Changing from one set of four values to the other changes the organization setting for Enable Cumulative Forecast Rollups in Setup. If this field is omitted, the setting isn't changed.
`forecastingDateType`	ForecastingDateType (enumeration of type string)	Required. The date type that forecast amounts are based on. Valid values are: • `OpportunityCloseDate` (default) • `ProductDate` • `ScheduleDate` • OLIMeasureCloseDateOnly • `ProductDateOnly` • `ScheduleDateOnly` • `OpportunityCustomDate` (Available in API version 57.0 and later) • `OLIMeasureOppCustomDateOnly` (Available in API version 57.0 and later)

Field Name	Field Type	Description
		Available in API version 42.0 and later. In API version 42.0 only, date types are read only and available only via API.
`forecastRangeSettings`	ForecastRangeSettings	Removed. The default periods and range selections in forecasts. In API version 53.0 and later, use globalForecastRangeSettings.
`hasProductFamily`	boolean	Required. Indicates whether the forecasting type has product family forecasts enabled. Available in API version 41.0 and later.
`isAmount`	boolean	Required. This read-only field indicates whether the forecast type is based on revenue amounts. The value of `isAmount` is always the opposite of the value of `isQuantity`.
`isAvailable`	boolean	Required. This read-only field indicates whether the forecast type can currently be used in the organization. For example, the revenue splits forecast type can't be used in an organization that doesn't have Opportunity Splits enabled.
`isQuantity`	boolean	Required. This read-only field indicates whether the forecast type is based on product quantities. The value of `isQuantity` is always the opposite of the value of `isAmount`.
`managerAdjustableCategoryApiNames`	string	This read-only field appears twice to specify the two forecast rollup categories that forecast managers can adjust in the organization for either cumulative forecast rollups or individual forecast category rollups. This field can only be used when the `enableAdjustments` field contains a value of true. If both the `managerAdjustableCategoryApiNames` and `ownerAdjustableCategoryApiNames` fields are being used, they must contain the same two values. Their values must also be consistent with the values of the `enableAdjustments` and `enableOwnerAdjustments` fields. Valid values for organizations using cumulative forecast rollups: • `bestcaseforecast` • `commitforecast` Valid values for organizations using individual forecast category rollups: • `bestcaseonly` • `commitonly`
`masterLabel`	string	Required. This read-only field indicates the UI label for the forecast type.
`name`	string	Required. The name of the forecast type. Each forecast type requires a specific string. Using ForecastingSettings, you can activate only these forecast types. • `LineItemQuantityProductDate`: Product Families - Quantity by product date. Available in API versions 47.0 and later.

Field Name	Field Type	Description
		• `LineItemQuantityScheduleDate`: Product Families - Quantity by schedule date. Available in API versions 47.0 and later. • `LineItemRevenueProductDate`: Product Families - Revenue by product date. Available in API versions 47.0 and later. • `LineItemRevenueScheduleDate`: Product Families - Revenue by schedule date. Available in API versions 47.0 and later. • `OpportunityLineItemQuantity`: Product Families - Quantity. • `OpportunityLineItemRevenue`: Product Families - Revenue. • `OpportunityOverlayRevenue`: Opportunity Overlay Splits - Revenue. • `OpportunityQuantity`: Opportunities - Quantity. • `OpportunityQuantityProductDate`: Opportunities - Quantity by product date. Available in API versions 43.0 and later. • `OpportunityQuantityScheduleDate`: Opportunities - Quantity by schedule date. Available in API versions 43.0 and later. • `OpportunityRevenue`: Opportunities - Revenue. • `OpportunityRevenueProductDate`: Opportunities - Revenue by product date. Available in API versions 43.0 and later. • `OpportunityRevenueScheduleDate`: Opportunities - Revenue by schedule date. Available in API versions 43.0 and later. • `OpportunitySplitRevenue`: Opportunity Revenue Splits - Revenue. • ***`Territory_Model_NameN`***: Territories, where *`Territory_Model_Name`* is the name of your active territory model in the API. *`Territory_Model_Name`* can be followed by *`N`*, an auto-generated number that distinguishes between territory forecast types. Available in API versions 44.0 and later. • ***`Territory_Model_NameN`***`_ProductFamily`: Deprecated. Territories - Product Families, where *`Territory_Model_Name`* is the name of your active territory model in the API and can be followed by *`N`*, an auto-generated number that distinguishes between territory forecast types. Available in API versions 45.0 and 46.0. For territory models created in API version 47.0 and later, ***`Territory_Model_NameN`*** is used. • The name of a custom opportunity split type that has been enabled as a forecast type. Custom split types are based on currency fields, which can contain revenue amounts only. To create and manage other forecast types in API version 52.0 and later, use ForecastingSourceDefinition, ForecastingType, and ForecastingTypeSource.

Field Name	Field Type	Description
opportunityListFieldsLabelMappings	OpportunityListFieldsLabelMapping	A read-only list of the API names and UI labels for all fields on the Opportunity object.
opportunityListFields SelectedSettings	OpportunityListFields SelectedSettings	Required. The fields selected to appear in the opportunity pane of the forecast page for the forecast type. One of the selected fields must be **Opportunity Name**. You can select up to 15 fields.
opportunityListFields UnselectedSettings	OpportunityListFields UnselectedSettings	Required. The fields *not* selected to appear in the opportunity pane of the forecast page for the forecast type.
opportunitySplitName	string	Indicates whether the forecasting type has a split type, and if so, the name of the split type. Available in API version 41.0 and later.
ownerAdjustableCategoryApiNames	string	This read-only field appears twice to specify the two forecast rollup categories that forecast owners can adjust in the organization, for either cumulative forecast rollups, or individual forecast category rollups. This field can only be used when the `enableOwnerAdjustments` field contains a value of true. If both the `managerAdjustableCategoryApiNames` and `ownerAdjustableCategoryApiNames` fields are being used, they must contain the same two values. Their values must also be consistent with the values of the `enableAdjustments` and `enableOwnerAdjustments` fields. Valid values for organizations using cumulative forecast rollups: • `bestcaseforecast` • `commitforecast` Valid values for organizations using individual forecast category rollups: • `bestcaseonly` • `commitonly`
quotasSettings	QuotasSettings	Removed. This field enables or disables the quota option in forecasts. In API version 53.0 and later, use globalQuotasSettings.
territory2ModelName	string	Indicates whether the forecasting type has a Territory2 model, and if so, the name of the Territory2 model. Available in API version 41.0 and later.

AdjustmentsSettings

The adjustment options for forecasts.

Field	Field Type	Description
allowExpandedColumns	boolean	Required. Set to `true` to show separate columns on the forecasts page for each adjustable forecast category and `false` to show adjustments when a user hovers over a forecast category. All forecast types must contain the same `allowExpandedColumns` value.

Field	Field Type	Description
`enableAdjustments`	boolean	Required. Set to `true` to enable manager adjustments and `false` to disable them. All forecast types must contain the same `enableAdjustments` value. Disabling adjustments results in adjustment data being purged.
`enableOwnerAdjustments`	boolean	Required. Set to `true` to enable owner adjustments for forecasts and `false` to disable them. All forecast types must contain the same `enableOwnerAdjustments` value. Disabling adjustments results in forecast adjustment data being purged.

ForecastingGroup

The group based on a custom picklist that is used to group or roll up forecast totals on the forecasts page. For example, group forecasts using a custom picklist for industry or sales type.

Field	Field Type	Description
`developerName`	string	Required. The API name that identifies the forecast group.
`forecastingGroupItems`	ForecastingGroupItem	Required. The picklist values for the forecast type. Possible values include the picklist values defined in `groupField`.
`groupField`	string	Required. The field name of the custom picklist used as a group. Possible values include custom, single-selection picklists available in `sourceObject`.
`masterLabel`	string	Required. This read-only field indicates the UI label for the forecast group.
`sourceObject`	string	Required. The source object for the picklist for the forecast group. Possible values include: • Opportunity • OpportunityLineItem • Product2

ForecastingGroupItem

The picklist value that is specified as the forecasting group for a forecast type, and the order it displays in on the forecasts page.

Field	Field Type	Description
`displayPosition`	int	Required. Indicates the display order of the values on the forecasts page.
`sourceAPIValue`	string	Required. The API name is that's derived from the group value.

ForecastRangeSettings

The default periods and range selections in forecasts. Users can forecast up to 15 months, 15 fiscal periods, or 8 quarters in the future or past. If your forecast range includes the current month, period, or quarter, the forecasts page shows the current month, period, or quarter by default. If not, the first month, period, or quarter of the range is selected. All forecast types must contain the same `forecastRangeSettings` field values.

Warning: If you change the time period from monthly to quarterly or quarterly to monthly, or you change the standard fiscal year, all adjustments, manager judgments, and quotas are purged. If you enable custom fiscal years, creating the first custom fiscal year deletes any quotas and adjustments in the corresponding and subsequent standard fiscal years. These changes trigger a forecast recalculation that can take significant time, depending on the quantity of your data.

Field	Field Type	Description
`beginning`	int	Required. Indicates the beginning month or quarter to display by default.
`displaying`	int	Required. Indicates the number of months or quarters to display by default. The maximum number of months is 12 and quarters is 8.
`periodType`	PeriodTypes (enumeration of type string)	Required. Indicates what type of period to use. Valid values are: • `Month` • `Quarter` • `Week` • `Year`

OpportunityListFieldsLabelMapping

A read-only list of the API names and UI labels for all fields on the Opportunity object.

Field	Field Type	Description
`field`	string	Required. The API name of the Opportunity field.
`label`	string	Required. The UI label of the Opportunity field.

OpportunityListFieldsSelectedSettings

The fields selected to appear in the opportunity pane of the forecast page for the forecast type. One of the selected fields must be **Opportunity Name**. You can select up to 15 fields.

Field	Field Type	Description
`field`	string	Specifies names of fields to display in the opportunity pane.

OpportunityListFieldsUnselectedSettings

The fields *not* selected to appear in the opportunity pane of the forecast page for the forecast type.

Field	Field Type	Description
field	string	Specifies names of fields not displayed in the opportunity pane.

QuotasSettings

QuotasSettings indicates if quotas are available in forecasts.

Field	Field Type	Description
showQuotas	boolean	Required. Set to true to enable quotas. All forecast types must contain the same showQuotas field value.

WeightedSourceCategory

This field can occur more than one time when specifying more than one forecast category to include in the rollup type. Each occurrence contains two subfields that specify a forecast category to include in the forecast rollup type and its weight. Some rollup types include more than one forecast category. This table shows the forecast categories that are included in each rollup type.

Field	Field Type	Description
sourceCategoryApiName	string	Required. Specifies the API name of a forecast category to include in the rollup type. The valid values are. • pipeline • best case • commit • closed • omitted • customcategory
weight	double	Required. Specifies the weight given to the forecast category when calculating the forecast for the rollup type. The only supported value is 1.0.

Declarative Metadata Sample Definition

The following is an example of a ForecastingSettings component that enables the Opportunity-Revenue forecast type, adjustments, owner adjustments, and quotas, and changes forecast range settings:

```
<?xml version="1.0" encoding="UTF-8"?>
<ForecastingSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <defaultToPersonalCurrency>false</defaultToPersonalCurrency>
    <enableForecasts>true</enableForecasts>
    <globalAdjustmentsSettings>
        <enableAdjustments>true</enableAdjustments>
        <enableOwnerAdjustments>true</enableOwnerAdjustments>
```

```
    </globalAdjustmentsSettings>
    <globalForecastRangeSettings>
        <beginning>1</beginning>
        <displaying>6</displaying>
        <periodType>Month</periodType>
    </globalForecastRangeSettings>
    <globalQuotasSettings>
        <showQuotas>true</showQuotas>
    </globalQuotasSettings>
    <forecastingCategoryMappings>
        <forecastingItemCategoryApiName>commitonly</forecastingItemCategoryApiName>
        <weightedSourceCategories>
            <sourceCategoryApiName>commit</sourceCategoryApiName>
            <weight>1.0</weight>
        </weightedSourceCategories>
    </forecastingCategoryMappings>
    <forecastingCategoryMappings>
        <forecastingItemCategoryApiName>closedonly</forecastingItemCategoryApiName>
        <weightedSourceCategories>
            <sourceCategoryApiName>closed</sourceCategoryApiName>
            <weight>1.0</weight>
        </weightedSourceCategories>
    </forecastingCategoryMappings>
    <forecastingCategoryMappings>
        <forecastingItemCategoryApiName>openpipeline</forecastingItemCategoryApiName>
        <weightedSourceCategories>
            <sourceCategoryApiName>most likely</sourceCategoryApiName>
            <weight>1.0</weight>
        </weightedSourceCategories>
        <weightedSourceCategories>
            <sourceCategoryApiName>commit</sourceCategoryApiName>
            <weight>1.0</weight>
        </weightedSourceCategories>
        <weightedSourceCategories>
            <sourceCategoryApiName>pipeline</sourceCategoryApiName>
            <weight>1.0</weight>
        </weightedSourceCategories>
        <weightedSourceCategories>
            <sourceCategoryApiName>best case</sourceCategoryApiName>
            <weight>1.0</weight>
        </weightedSourceCategories>
    </forecastingCategoryMappings>
    <forecastingCategoryMappings>
        <forecastingItemCategoryApiName>omittedonly</forecastingItemCategoryApiName>
        <weightedSourceCategories>
            <sourceCategoryApiName>omitted</sourceCategoryApiName>
            <weight>1.0</weight>
        </weightedSourceCategories>
    </forecastingCategoryMappings>
    <forecastingCategoryMappings>
        <forecastingItemCategoryApiName>bestcaseforecast</forecastingItemCategoryApiName>

        <weightedSourceCategories>
            <sourceCategoryApiName>most likely</sourceCategoryApiName>
```

```
            <weight>1.0</weight>
        </weightedSourceCategories>
        <weightedSourceCategories>
            <sourceCategoryApiName>commit</sourceCategoryApiName>
            <weight>1.0</weight>
        </weightedSourceCategories>
        <weightedSourceCategories>
            <sourceCategoryApiName>closed</sourceCategoryApiName>
            <weight>1.0</weight>
        </weightedSourceCategories>
        <weightedSourceCategories>
            <sourceCategoryApiName>best case</sourceCategoryApiName>
            <weight>1.0</weight>
        </weightedSourceCategories>
    </forecastingCategoryMappings>
    <forecastingCategoryMappings>
        <forecastingItemCategoryApiName>pipelineonly</forecastingItemCategoryApiName>
        <weightedSourceCategories>
            <sourceCategoryApiName>pipeline</sourceCategoryApiName>
            <weight>1.0</weight>
        </weightedSourceCategories>
    </forecastingCategoryMappings>
    <forecastingCategoryMappings>
        <forecastingItemCategoryApiName>commitforecast</forecastingItemCategoryApiName>
        <weightedSourceCategories>
            <sourceCategoryApiName>closed</sourceCategoryApiName>
            <weight>1.0</weight>
        </weightedSourceCategories>
        <weightedSourceCategories>
            <sourceCategoryApiName>commit</sourceCategoryApiName>
            <weight>1.0</weight>
        </weightedSourceCategories>
    </forecastingCategoryMappings>
    <forecastingCategoryMappings>
        <forecastingItemCategoryApiName>bestcaseonly</forecastingItemCategoryApiName>
        <weightedSourceCategories>
            <sourceCategoryApiName>best case</sourceCategoryApiName>
            <weight>1.0</weight>
        </weightedSourceCategories>
    </forecastingCategoryMappings>
    <forecastingTypeSettings>
<name>OpportunityRevenue</name>
        <active>true</active>
        <hasProductFamily>false</hasProductFamily>
        <isAmount>true</isAmount>
        <isAvailable>true</isAvailable>
        <isQuantity>false</isQuantity>
        <managerAdjustableCategoryApiNames>commitonly</managerAdjustableCategoryApiNames>

      <managerAdjustableCategoryApiNames>bestcaseonly</managerAdjustableCategoryApiNames>

        <masterLabel>Opportunities</masterLabel>
<displayedCategoryApiNames>closedonly</displayedCategoryApiNames>
        <displayedCategoryApiNames>commitonly</displayedCategoryApiNames>
```

```
        <displayedCategoryApiNames>bestcaseonly</displayedCategoryApiNames>
        <displayedCategoryApiNames>pipelineonly</displayedCategoryApiNames>
        <forecastedCategoryApiNames>commitonly</forecastedCategoryApiNames>
        <forecastedCategoryApiNames>closedonly</forecastedCategoryApiNames>
        <forecastedCategoryApiNames>pipelineonly</forecastedCategoryApiNames>
        <forecastedCategoryApiNames>bestcaseonly</forecastedCategoryApiNames>
        <forecastingDateType>OpportunityCloseDate</forecastingDateType>
        <opportunityListFieldsSelectedSettings>
            <field>OPPORTUNITY.NAME</field>
        </opportunityListFieldsSelectedSettings>
    </forecastingTypeSettings>
</ForecastingSettings>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

HighVelocitySalesSettings

Represents an org's Sales Engagement settings. With Sales Engagement, you can make your inside sales team as effective as possible.

This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

HighVelocitySalesSettings values are stored in a single file named `HighVelocitySales.settings` in the `settings` directory of the corresponding package directory.

Version

HighVelocitySalesSettings components are available in API version 47.0 and later.

Fields

Field Name	Field Type	Description
`enableACAutoSendEmail`	boolean	Indicates whether a cadence step of type Automated Send can be created `(true)` or not `(false)`. Allowing Salesforce to automatically send an email to a prospect can make your sales team more efficient, because reps don't have to send the email themselves. Default value is `true`. Available in API version 48.0 and later.

Field Name	Field Type	Description
enableACChangeTargetAssignee	boolean	Indicates whether target assignees, target owners, and users with access to related cadences can update target assignees (true) or not (false). Available in API version 50.0 and later.
enableACSkipWeekends	boolean	Indicates whether Skip Weekends is enabled (true) or not (false). Available in API version 48.0 and later.
enableBusinessHours	boolean	Indicates whether Business Hours is enabled in Sales Engagement (true) or not (false). Available in API version 58.0 and later.
enableCadenceVariantTestingPref	boolean	Indicates whether AB testing for cadence steps is enabled (true) or not (false). Available in API version 53.0 and later.
enableChgTgtAssigneeUsrPermPref	boolean	Indicates whether the change target assignee action is controlled by user permission (true) or not (false). Available in API version 51.0 and later.
enableDispositionCategory	boolean	Indicates whether Call Outcomes For Branching is enabled in Sales Engagement (true) or not (false). Use Call Outcomes For Branching to group calls into different outcome categories such as "Left Voicemail" or "Not Interested." You can see the outcomes in a report, or use them to determine how cadences are branched. enableHighVelocitySales must be true to use Sales Engagement. Default value is false.
enableEngagementWaveAnalyticsPref	boolean	Indicates whether you can see engagement statistics in CRM Analytics (true) or not (false). Use CRM Analytics to analyze information about calls, engagement, and how each sales rep moves through their cadence steps.
enableHighVelocitySales	boolean	Indicates whether Sales Engagement is enabled (true) or not (false). If enabled, it turns on the features required for the product and makes the app available to users. Default value is false.
enableHighVelocitySalesSetup	boolean	Indicates whether Sales Engagement is enabled (true) or not (false). Default value is false.

Field Name	Field Type	Description
enableInvoiceAttributionPref	boolean	Indicates whether Invoice Attribution is enabled (true) or not (false). Available in API version 56.0 and later.
enableLogACallForCTIPref	boolean	Indicates whether Log a Call appears to CTI users by default (true) or not (false). Available in API version 54.0 and later.
enableLogTasksForLinkedInPref	boolean	Indicates whether users can log standard tasks upon completion of a LinkedIn step (true) or not (false). Available in API version 54.0 and later.
enableMultipleCadencesPref	boolean	Indicates whether targets can be assigned to multiple cadences (true) or not (false). Available in API version 57.0 and later.
enableOpportunityAttributionPermPref	boolean	Indicates whether Opportunity Attribution is enabled (true) or not (false). Available in API version 51.0 and later.
enableQuickCadenceAutoSendEmail	boolean	Indicates whether Automated Email send is enabled for Sales Engagement quick cadences (true) or not (false). Available in API version 57.0 and later.
enableTaskLoggingPref	boolean	Indicates whether users can log tasks after manual completion of a cadence step (true) or not (false). Available in API version 56.0 and later.

Declarative Metadata Sample Definition

The following is an example of the HighVelocitySales.settings file:

```
<?xml version="1.0" encoding="UTF-8"?>
<HighVelocitySalesSettings xmlns="http://soap.sforce.com/2006/04/metadata">
  <enableACAutoSendEmail>false</enableACAutoSendEmail>
  <enableACChangeTargetAssignee>false</enableACChangeTargetAssignee>
  <enableACSkipWeekends>false</enableACSkipWeekends>
  <enableBusinessHours>false</enableBusinessHours>
  <enableCadenceVariantTestingPref>false</enableCadenceVariantTestingPref>
  <enableChgTgtAssigneeUsrPermPref>false</enableChgTgtAssigneeUsrPermPref>
  <enableDispositionCategory>true</enableDispositionCategory>
  <enableEngagementWaveAnalyticsPref>true</enableEngagementWaveAnalyticsPref>
  <enableHighVelocitySales>true</enableHighVelocitySales>
  <enableHighVelocitySalesSetup>true</enableHighVelocitySalesSetup>
  <enableInvoiceAttributionPref>false</enableInvoiceAttributionPref>
  <enableLogACallForCTIPref>false</enableLogACallForCTIPref>
```

```
  <enableLogTasksForLinkedInPref>false</enableLogTasksForLinkedInPref>
  <enableMultipleCadencesPref>false</enableMultipleCadencesPref>
  <enableOpportunityAttributionPermPref>false</enableOpportunityAttributionPermPref>
  <enableQuickCadenceAutoSendEmail>false</enableQuickCadenceAutoSendEmail>
  <enableTaskLoggingPref>true</enableTaskLoggingPref>
</HighVelocitySalesSettings>
```

Example Package Manifest

The following is an example package manifest used to deploy or retrieve the HighVelocitySalesSettings settings metadata:

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
 <types>
  <members>HighVelocitySalesSettings</members>
  <name>Settings</name>
 </types>
 <version>47.0</version>
</Package>
```

IdeasSettings

Represents the metadata used to manage settings for Ideas.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

IdeasSettings is stored in one file named `Ideas.settings` in the `settings` folder of the corresponding package directory. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

IdeasSettings is available in API version 27.0 and later.

Ideas

Represents settings for Ideas and Idea Themes.

Fields

Field Name	Field Type	Description
`enableIdeaThemes`	boolean	Indicates whether Idea Themes is enabled (`true`) or not (`false`).
`enableIdeas`	boolean	Indicates whether Ideas is enabled (`true`) or not (`false`).
`enableIdeasReputation`	boolean	Indicates whether Reputation is enabled (`true`) or not (`false`). You can't enable IdeasReputation without enabling the Ideas Reputation

Field Name	Field Type	Description
		permission in your organization. This field is available in API version 28.0 and later.
`enableChatterProfile`	boolean	Indicates that the Chatter user profile is used for Ideas user profiles. If `enableChatterProfile` is `true`, the `ideasProfilePage` value must not be specified. If `enableChatterProfile` is `false`, then specify a `ideasProfilePage` value, otherwise the Ideas zone profile is used. This field is available in API version 29.0 and later.
`ideasProfilePage`	string	The name of the Visualforce page to use for a custom Ideas user profile, if `enableChatterProfile` is `false`. If `enableChatterProfile` is `false`, then specify a `ideasProfilePage` value, otherwise the Ideas zone profile is used. This field is available in API version 29.0 and later.
`halfLife`	double	Indicates how quickly old ideas drop in ranking on the Popular Ideas subtab. The half-life setting determines how the number of days after which old ideas drop in ranking on the Popular Ideas subtab, to make room for ideas with more recent votes. A shorter half-life moves older ideas down the page faster than a longer half-life.

Declarative Metadata Sample Definition

The following is an example `ideas.settings` metadata file:

```
<?xml version="1.0" encoding="UTF-8"?>
<IdeasSettings xmlns="http://soap.sforce.com/2006/04/metadata">
   <enableIdeaThemes>true</enableIdeaThemes>
   <enableIdeas>true</enableIdeas>
   <enableIdeasReputation>true</enableIdeasReputation>
   <enableChatterProfile>false</enableChatterProfile>
   <ideasProfilePage>name of Visualforce page</ideasProfilePage>
   <halfLife>2.6</halfLife>
</IdeasSettings>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

IdentityProviderSettings

Represents the settings used to enable or disable Salesforce as a SAML identity provider for single sign-on (SSO).

Parent Type and Manifest Access

This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all the settings metadata types for the org are accessed using the "Settings" name. See Settings for more details.

File Suffix and Directory Location

`IdentityProviderSettings` values are stored in the `IdentityProvider.settings` file in the `settings` folder. The `.settings` files are different from other named components, because there is only one settings file for each settings component.

Version

IdentityProviderSettings components are available in API version 57.0 and later.

Special Access Rules

To access IdentityProviderSettings, a user must have the Customize Application user permission.

Fields

Field Name	Description
`certificateName`	**Field Type** string **Description** Required. The certificate that Salesforce uses to communicate with SAML SSO service providers, such as third-party service providers or another Salesforce org acting as a service provider. You can enter the name of a self-signed certificate or a certificate signed by a certificate authority.
`enableIdentityProvider`	**Field Type** boolean **Description** Required. Indicates whether Salesforce can be used as a SAML identity provider (`true`) or not (`false`).

Declarative Metadata Sample Definition

The following is an example of an IdentityProviderSettings component. In this example, Salesforce is enabled as a SAML identity provider.

```
<?xml version="1.0" encoding="UTF-8"?>
<IdentityProviderSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <certificateName>Certificate Name</certificateName>
    <enableIdentityProvider>true</enableIdentityProvider>
</IdentityProviderSettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>IdentityProvider</members>
        <name>Settings</name>
    </types>
    <version>57.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type doesn't support the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

IframeWhiteListUrlSettings

Represents settings related to the list of trusted external domains that you allow to frame your Visualforce pages or surveys. This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. Because changing terms in our code can break current implementations, we maintained this metadata type's name.

File Suffix and Directory Location

IframeWhiteListUrlSettings on page 1973 values are stored in the IframeWhiteListUrlSettings on page 1973 `.settings` file in the `settings` directory. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

IframeWhiteListUrlSettings on page 1973 components are available in API version 49.0 and later.

Fields

Field Name	Field Type	Description
`iframeWhiteListUrls`	IframeWhiteListUrl[]	The list of external domains that you allow to frame your Visualforce pages or surveys.

IframeWhiteListUrl

Represents the external domains that you allow to frame your Visualforce pages or surveys.

Field Name	Field Type	Description
`context`	IFrameWhitelistContext (enumeration of type string)	Required. The type of content in the iframe. Valid values are: • `Surveys` • `VisualforcePages` • `DisclosureAndComplianceHubConnector`
`url`	string	The unique domain that is allowed to frame your Visualforce pages, surveys, or Disclosure and Compliance Hub Connector. Accepts these formats: `example.com`, `*example.com`, and `https://example.com`.

Declarative Metadata Sample Definition

The following is an example of a IframeWhiteListUrlSettings on page 1973 component.

```
<?xml version="1.0" encoding="UTF-8"?>
<IframeWhiteListUrlSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <iframeWhiteListUrl>
      <context>Surveys></context>
      <url>example1.com</url>
    </iframeWhiteListUrl>
    <iframeWhiteListUrl>
      <context>VisualforcePages</context>
      <url>example2.com</url>
     </iframeWhiteListUrl>
     <iframeWhiteListUrl>
        <context>DisclosureAndComplianceHubConnector</context>
        <url>example3.com</url>
     </iframeWhiteListUrl>
</IframeWhiteListUrlSettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
      <members>IframeWhiteListUrlSettings</members>
      <name>IframeWhiteListUrlSettings</name>
   </types>
    <version>62.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type doesn't support the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

IncidentMgmtSettings

Represents settings for Customer Service Incident Management and Broadcast Communications.

Parent Type and Manifest Access

This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all the settings metadata types for the org are accessed using the "Settings" name. See Settings for more details.

File Suffix and Directory Location

`IncidentMgmtSettings` values are stored in the `IncidentMgmt.settings` file in the `settings` folder. The `.settings` files are different from other named components, because there is only one settings file for each settings component.

Version

IncidentMgmtSettings components are available in API version 54.0 and later.

Special Access Rules

IncidentMgmtSettings requires a Service Cloud license.

Fields

Field Name	Description
`enableAlertBroadcastType`	**Field Type** boolean **Description** Indicates whether broadcast communication alerts are enabled (`true`) or not (`false`). Lets incident managers send disruptive in-app notifications when an incident occurs. Default is `false`. Available in API version 57.0 and later.
`enableEmailBroadcastType`	**Field Type** boolean **Description** Indicates whether broadcast communication emails are enabled (`true`) or not (`false`). Lets users send an email with critical information to impacted customers. Default is `false`. Available in API version 56.0 and later.
`enableIncidentMgmt`	**Field Type** boolean **Description** Indicates whether Customer Service Incident Management is enabled (`true`) or not (`false`). Customer Service Incident Management is a Service Cloud solution that helps your teams track large-scale disruptions and delegate tasks to the right experts to ensure that your business delivers on customer expectations. Default is `true`. Available in API version 54.0 and later.

Field Name	Description
enableSiteBannerBroadcastType	**Field Type** boolean **Description** Indicates whether broadcast communication for site banners is enabled (`true`) or not (`false`). Lets users add a banner with critical information to your Aura and LWR sites. Default is `false`. Available in API version 56.0 and later.
enableSlackBroadcastType	**Field Type** boolean **Description** Indicates whether broadcast communication Slack messages are enabled (`true`) or not (`false`). Lets incident managers send broadcasts to Slack when an incident occurs. Default is `false`. Available in API version 57.0 and later.

Declarative Metadata Sample Definition

The following is an example of an IncidentMgmtSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<AccountSettings xmlns="http://soap.sforce.com/2006/04/metadata">
 <enableAccountDiscovery>true</enableAccountDiscovery>
 <enableAccountHistoryTracking>true</enableAccountHistoryTracking>
 <enableAccountInsightsInMobile>false</enableAccountInsightsInMobile>
 <enableAccountOwnerReport>true</enableAccountOwnerReport>
 <enableAccountTeams>true</enableAccountTeams>
 <enableContactHistoryTracking>true</enableContactHistoryTracking>
 <enableRelateContactToMultipleAccounts>true</enableRelateContactToMultipleAccounts>
 <showViewHierarchyLink>true</showViewHierarchyLink>
</AccountSettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>IncidentMgmt</members>
        <name>Settings</name>
    </types>
    <version>54.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

IndustriesEinsteinFeatureSettings

Represents the settings for enabling the Industries Einstein feature.

This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for more details.

File Suffix and Directory Location

IndustriesEinsteinFeatureSettings values are stored in a single file named `IndustriesEinsteinFeature.settings` in the `settings` folder. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

IndustriesEinsteinFeatureSettings components are available in API version 57.0 and later.

Fields

Field Name	Description
`documentReaderConfidenceOrgValue`	**Field Type** double **Description** Required. Specify the confidence score threshold to indicate the reliability of data in a document. You can enter a number from 0 to 100, with up to two decimal places, where 0 is the least confident and 100 is the most confident.

Declarative Metadata Sample Definition

The following is an example of a IndustriesEinsteinFeatureSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<IndustriesEinsteinFeatureSettings xmlns="http://soap.sforce.com/2006/04/metadata">
   <documentReaderConfidenceOrgValue>61</documentReaderConfidenceOrgValue>
</IndustriesEinsteinFeatureSettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
  <types>
    <members>IndustriesEinsteinFeature</members>
    <name>Settings</name>
  </types>
  <version>64.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

IndustriesLoyaltySettings

Represents the settings to enable capabilities of Loyalty Management.

This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for more details.

File Suffix and Directory Location

IndustriesLoyaltySettings values are stored in a single file named `IndustriesLoyalty.settings` in the `settings` folder. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

IndustriesLoyaltySettings components are available in API version 53.0 and later.

Fields

Field Name	Field Type	Description
enableAutomaticVoucherCodeGeneration	boolean	Indicates whether the capability to automatically generate voucher codes is enabled (`true`) or not (`false`). This field is available in API version 57.0 and later.
enableNQPRealTimePointBalance	boolean	Indicates whether the capability to update the non-qualifying point balance of members in real time is enabled (`true`) or disabled (`false`) for your org. The default value is `true`.
enableFixedTypeNQPAggregation	boolean	Indicates whether the capability to aggregate and expire fixed type non-qualifying points in batches is enabled (`true`) or disabled (`false`) for your org. The default value is `true`. This field is available in API version 54.0 or later.
enableNonQualifyingPointsConsolidation	boolean	Indicates whether the capability to aggregate and expire fixed type non-qualifying points in real time is enabled (`true`) or disabled (`false`) for your org. The default value is `true`. This field is available in API version 58.0 or later.
enableQPRealTimePointBalance	boolean	Indicates whether the real time qualifying points balance update capability is enabled (`true`) or disabled (`false`) for your org. The default value is `true`. This field is available in API version 55.0 or later.

Field Name	Field Type	Description
enableLoyaltyRulesVerifyObjMemberSegment	boolean	Indicates whether the capability that allows: • Loyalty program process rules to process transaction journals only when the loyalty program member is part of a Data Cloud segment associated with the rule's promotion is enabled (`true`) or disabled (`false`) for your org. • The Promotion Eligibility component on Loyalty Program Member record page to categorize promotions based on whether the member belongs to the promotion's campaign or Data Cloud segment is enabled (`true`) or disabled (`false`) for your org. The default value is `false`. This field is available in API version 55.0 or later.
enableLoyaltyRedeemedPointsExpirationInfoPref	boolean	Indicates whether the capability that automatically calculates and adds the expiration date of points credited back to members for canceled redemptions is enabled (`true`) or disabled (`false`) for your org. The default value is `true`. This field is available in API version 55.0 or later.
enableLoyaltyServiceExcellence	boolean	Indicates whether Service Console for Loyalty Management is enabled (`true`) or disabled (`false`) for your org. The default value is `false`. This field is available in API version 57.0 or later.
enableLoyaltyApiAccessForExternalSiteUsers	boolean	Indicates whether the capability that allows customers to join and leave loyalty programs and allows members to join and opt out of promotions from Experience Cloud sites is enabled (`true`) or disabled (`false`) for your org. The default value is `false`. This field is available in API version 58.0 or later.
enableAutomaticMemberTierAssessmentSelection	boolean	Indicates whether the capability that automatically selects members as eligible for tier assessment when members' qualifying points balance changes is enabled (`true`) or disabled (`false`) for your org. The default value is `false`. This field is available in API version 58.0 or later.
enablePointsLifecycleTracking	boolean	Indicates whether the capability that allows the company to trace how members redeem their accrued fixed-model non-qualifying points is enabled (`true`) or disabled (`false`) for your org. The default value is `false`. This field is available in API version 62.0 or later.
enableNegativePointBalance	boolean	Indicates whether the capability that lets loyalty program members to hold negative point balances for non-qualifying currencies is enabled (`true`) or disabled (`false`) for your org. The default value is `false`. This field is available in API version 61.0 or later.
enableSegmentQueryByDataGraph	boolean	Indicates whether the capability, which uses Data Cloud data graphs to query for Data Cloud segments that customers are part of, is enabled (`true`) or disabled (`false`). The default value is `false`. This field is available in API version 62.0 or later.
enablePromSetupProdRuleStatusInheritDplymt	boolean	Indicates whether the capability, which allows Promotion Setup processes and rules to be deployed to target orgs in the same status as the source

Field Name	Field Type	Description
		orgs when using the LoyaltyProgramSetup metadata type, is enabled (`true`) or disabled (`false`). The default value is `false`. This field is available in API version 62.0 or later.
`enableLoyaltyGenerativeAi`	boolean	Indicates whether the Einstein AI capabilities of Loyalty Management are enabled (`true`) or not (`false`). The default value is `false`. This field is available in API version 62.0 or later.
enableTransferPointsToMemberGroupsRealtime	boolean	Indicates whether the capability that automates real-time transfer of a member's non-qualifying points to associated group is enabled (`true`) or not (`false`). The default value is `false`. This field is available in API version 64.0 or later.
enableUsePromPtyUsageForEngmtTrail	boolean	Indicates whether the capability that uses Promotion Party Usage records to store attributes for Engagement Trail promotions is enabled (`true`) or not (`false`). The default value is `true`. This field is available in API version 64.0 or later.
enableSegmentQueryApiMultipleDataSpace	boolean	Indicates whether the capability that allows Query API to verify the Data Cloud segments that customers are part of across multiple data spaces is enabled (`true`) or not (`false`). The default value is `false`. This field is available in API version 64.0 or later.
enableConfigureClubs	boolean	Indicates whether the capability to create and manage clubs is enabled (`true`) or not (`false`). The default value is `false`. This field is available in API version 64.0 or later.

Declarative Metadata Sample Definition

The following is an example of a IndustriesLoyaltySettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<IndustriesLoyaltySettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <enableNQPRealTimePointBalance>true</enableNQPRealTimePointBalance>
    <enableFixedTypeNQPAggregation>true</enableFixedTypeNQPAggregation>
    <enableQPRealTimePointBalance>true</enableQPRealTimePointBalance>

<enableLoyaltyRulesVerifyCdpMemberSegment>false</enableLoyaltyRulesVerifyCdpMemberSegment>


<enableLoyaltyRedeemedPointsExpirationInfoPref>true</enableLoyaltyRedeemedPointsExpirationInfoPref>

    <enableLoyaltyServiceExcellence>true</enableLoyaltyServiceExcellence>
    <enableAutomaticVoucherCodeGeneration>true</enableAutomaticVoucherCodeGeneration>

<enableLoyaltyApiAccessForExternalSiteUsers>false</enableLoyaltyApiAccessForExternalSiteUsers>


<enableAutomaticMemberTierAssessmentSelection>false</enableAutomaticMemberTierAssessmentSelection>
```

```
    <enableNonQualifyingPointsConsolidation>false</enableNonQualifyingPointsConsolidation>

    <enablePointsLifecycleTracking>false</enablePointsLifecycleTracking>
    <enableNegativePointBalance>false</enableNegativePointBalance>
    <enableSegmentQueryByDataGraph>false</enableSegmentQueryByDataGraph>

<enablePromSetupProcRuleStatusInheritDplymt>false</enablePromSetupProcRuleStatusInheritDplymt>

    <enableLoyaltyGenerativeAi>false</enableLoyaltyGenerativeAi>

<enableTransferPointsToMemberGroupsRealtime>false</enableTransferPointsToMemberGroupsRealtime>

    <enableUsePromPtyUsageForEngmtTrail>false</enableUsePromPtyUsageForEngmtTrail>
    <enableSegmentQueryApiMultipleDataSpace>true</enableSegmentQueryApiMultipleDataSpace>

    <enableConfigureClubs>true</enableConfigureClubs>
</IndustriesLoyaltySettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
  <types>
    <members>IndustriesLoyalty</members>
    <name>Settings</name>
  </types>
  <version>53.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

IndustriesSettings

Represents settings for industries verticals such as Financial Services Cloud, Consumer Goods Cloud, Public Sector Solutions, Education Cloud, Salesforce Scheduler, Life Sciences Cloud, and Health Cloud.

This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. Because changing terms in our code can break current implementations, we maintained this metadata type's name.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for more details.

File Suffix and Directory Location

IndustriesSettings are stored in a single file named `Industries.settings` in the `settings` directory.

Version

Industries settings are available in API version 47.0 and later.

Special Access Rules

Settings are specific to an industry vertical and are only available to customers with org editions where the vertical is enabled.

Fields for Financial Services Cloud

Users need the FSC Insurance permission set to access the settings available in editions with Financial Services Cloud.

Field Name	Field Type	Description
`allowMultipleProducersToWorkOnSamePolicy`	boolean	Indicates whether multiple producers can be assigned to the same insurance policy. The default is `false`. This field is available in editions where Financial Services Cloud is enabled.
`createCustomerPropertyFromLAProperty`	boolean	When importing loan application data, indicates whether to create a customer property record from a loan application property record to represent the new home.
`createFinancialAccountFromLAAsset`	boolean	When importing loan application data, indicates whether to create financial account records from the assets listed in the loan application.
`createFinancialAccountFromLALiability`	boolean	When importing loan application data, indicates whether to create financial account records from the liabilities listed in the loan application.
`createFinancialAccountsFromLAFinancials`	boolean	When importing loan application data, indicates whether to create a financial account record that represents the mortgage loan.
`createFinancialAccountsFromLAProperty`	boolean	When importing loan application data, indicates whether to create a financial account record from the loan application property to represent the new home.
`createFSCAssetFromLAAsset`	boolean	When importing loan application data, indicates whether to create asset records from the assets listed in the loan application.
`createFSCAssetFromLAProperty`	boolean	When importing loan application data, indicates whether to create an asset record from a loan application property record to represent the new home that was acquired.
`createFSCLiabilityFromLAFinancial`	boolean	When importing loan application data, indicates whether to create a liability record from a loan application property record to represent the new mortgage loan.
`createFSCLiabilityFromLALiability`	boolean	When importing loan application data, indicates whether to create liability records from the liabilities listed in the loan application.
`enableAccessToMasterListOfCoverageTypes`	boolean	Indicates whether insurance agents can access the main list of coverage types. The default is `false`. This field is available in editions where Financial Services Cloud is enabled.

Field Name	Field Type	Description
enableCallReportAdminContextPref	boolean	Indicates whether admins can configure the Account Interactions component and use a component attribute to control whether end users can view data on the component with admin privileges (true) or not (false). The default is false.
enableCalculationUsingParentPolicyOnly	boolean	Indicates whether the policy premiums are calculated by using only the parent policy's premium (true) or not (false). Use this for hierarchical policies where premiums are stored at different levels. The default value is false.
enableCompliantDataSharingForInteraction	boolean	Indicates whether compliant data sharing is enabled for the Interaction object (true) or not (false). The default value is false.
enableClaimMgmt	boolean	Indicates whether the recording and processing of the first loss notifications and managing claim participants, coverages, and settlements are enabled (true) or not (false). The default value is false.
enableCompliantDataSharingForAccount	boolean	Indicates whether the Compliant Data Sharing feature is enabled for the Account object. The default is false. This field is available in editions where Financial Services Cloud is enabled.
enableCompliantDataSharingForCustomObjects	boolean	Indicates whether the Compliant Data Sharing feature is enabled for custom objects (true) or not (false). The default value is false. This field is available in editions where Financial Services Cloud is enabled.
enableCompliantDataSharingForOpportunity	boolean	Indicates whether the Compliant Data Sharing feature is enabled for the Opportunity object (true) or not (false). The default value is false. This field is available in editions where Financial Services Cloud is enabled.
enableCompliantDataSharingForInteractionSummary	boolean	Indicates whether Compliant Data Sharing is enabled for the Interaction Summary object (true) or not (false). The default value is false.
enableDealManagement	boolean	Indicates whether the Financial Deal Management feature is enabled. The default is false. This field is available in editions where Financial Services Cloud is enabled.
enableDiscoveryFrameworkMetadata	boolean	Indicates whether the Assessment Question and Assessment Question Set features of the Discovery Framework Metadata are enabled (true) or not (false). The default value is false.
enableEinsteinDocReaderMappings	boolean	Indicates whether the Intelligent Form Reader Mappings feature is enabled. The default is false.
enableEnhancedQuestionCreation	boolean	Indicates whether the Enhanced Question Creation Experience feature of the Discovery Framework is enabled true) or not (false). The default value is false.

Field Name	Field Type	Description
`enableFinancialAccountMgmt`	boolean	Indicates whether the Financial Account Management Standard Objects setting is enabled for your org. The default is `false`. This field is available in editions where Financial Services Cloud is enabled.
`enableFinancialDealRoleHierarchy`	boolean	Indicates whether data sharing for the financial deals is configured to follow the role-based hierarchy (`true`) or not (`false`). The default value is `false`.
`enableFinancialDealCallReportCmpPref`	boolean	Indicates whether admins can use junction objects between the Financial Deal object and the Interaction and Interaction Summary objects within the Interaction Summary/Sharing component (`true`) or not (`false`). This field is available in API version 54.0 and later.
`enableFinancialDealCallReportPref`	boolean	Indicates whether the junction object is enabled between the Financial Deal object and the Interaction and Interaction Summary objects (`true`) or not (`false`). This field is available in API version 54.0 and later.
`enableFSCInsuranceReport`	boolean	Indicates whether sales managers can access the dashboard and prebuilt reports. The default is `false`. This field is available in API version 48.0 and later in editions where Financial Services Cloud is enabled. Note: This setting can be enabled only if the `allowMultipleProducersToWorkOnSamePolicy` setting is already set to `true`.
`enableIndustriesAssessment`	boolean	Indicates whether the Industries Assessment feature of the Discovery Framework is enabled `true`) or not (`false`). The default value is `false`.
`enableIndustriesKYC`	boolean	Indicates whether the Industries KYC (Know Your Customer) is enabled `true`) or not (`false`). The default value is `false`.
`enableInteractionRoleHierarchy`	boolean	Indicates whether role hierarchy-based sharing is available for interactions to users in the organization `true`) or not (`false`). The default value is `false`.
`enableInteractionSummaryPref`	boolean	Indicates whether the Interaction Summary setting is enabled for your org. The default is `false`. This field is available in editions where Financial Services Cloud is enabled.
`enableInteractionSummaryRoleHierarchy`	boolean	Indicates whether the Role-Hierarchy-Based Sharing for Interaction Summaries is enabled for your org. The default is `false`. This field is available in editions where Financial Services Cloud is enabled.
`enableManyToManyRelationships`	boolean	Indicates whether insurance can manage many-to-many relationships between claims and cases, claims and assets, and assets and policy participants. For example, if set to `true`, agents can handle multiple claims through one case or have multiple cases handling one claim.

Field Name	Field Type	Description
		The default is `false`. This field is available in editions where Financial Services Cloud is enabled.
`enableMortgageRlaTotalsOrgPref`	boolean	Indicates whether the calculation of assets and liabilities for residential loan application records is enabled for your org (true) or not (false). The default is `false`. This field is available in editions where Financial Services Cloud is enabled.
`enablePolicyAdministration`	boolean	Indicates whether the data model related to policy administrator is enabled `true`) or not (`false`). If this option is enabled, entities such as transactions and transaction details are available within the policy admin data model. The default value is `false`.
`enableRBLUsingCalcService`	boolean	Indicates whether Roll-by-Lookup (RBL) Using Calc Service is enabled for your org. The default is `false`. This field is available in editions where Financial Services Cloud is enabled.
`enableRecordRollup`	boolean	Indicates whether Record Rollup Optimization is enabled for your org. The default is `false`. This field is available in editions where Financial Services Cloud is enabled.
`enableReferralScoring`	boolean	Indicates whether Einstein Referral Scoring for Financial Services Cloud is enabled for your org (true) or not (false). The default is `false`. This field is available in editions where Financial Services Cloud is enabled.
`enableSlackForCib`	boolean	Indicates whether the access to Financial Services Cloud capabilities on Slack is enabled (`true`) or not (`false`). The default value is `false`.
`enableSyncInteractionsPref`	boolean	Indicates whether the Sync Interactions with Einstein Activity Capture setting is enabled (`true`) or not (`false`). The default value is `false`.
`enableTimelinePref`	boolean	Indicates whether the Timeline features are enabled for your organization (`true`) or not (`false`). The default value is `false`.
`loanApplicantAddressAutoCreation`	boolean	Indicates whether automatic generation of loan applicant records for new residential loan applications that are associated with person accounts is enabled for your org. The default is `false`. This field is available in editions where Financial Services Cloud is enabled. Available in API version 51.0 and later.
`loanApplicantAutoCreation`	boolean	Indicates whether automatic generation of loan applicant address records for new residential loan applications that are associated with person accounts is enabled for your org. The default is `false`. This field is available in editions where Financial Services Cloud is enabled. Available in API version 51.0 and later.
`rlaEditIfAccHasEdit`	boolean	Indicates whether a user can edit a residential loan application only if they have edit access on the account (`true`) or not (`false`). The default value is `false`.

Field Name	Field Type	Description
`transformRBLtoDPE`	boolean	Indicates whether you can convert RBL rules into Data Processing Engine definitions for faster calculations. The default is `false`.

Fields for Health Cloud

Field Name	Field Type	Description
`enableAuthorizationCustomSharingPCU`	boolean	Indicates whether to enable custom sharing to give your users access to view and manage electronic consent forms. Users with a Customer Community Plus license can share Authorization Form Texts and Data Use Purpose records with Accounts, Contracts, and Users specified in the Information Authorization Request record.
`enableCandidateMatching`	boolean	Indicates whether to enable Candidate Matching to automatically match candidates to clinical trials.
`enableCareMgmtSlackAccess`	boolean	Indicates whether Care Coordination for Slack app is enabled for your org (`true`) or not (`false`). Available in API version 56.0 and later.
`enableClinicalDataModel`	boolean	Indicates whether Clinical Data Model is enabled for your org (`true`) or not (`false`). The default is `false`. Available in API version 51.0 and later.
`enableContactCenterAccess`	boolean	Indicates whether Contact Center for Health Cloud app is enabled for your org (`true`) or not (`false`). The default is `false`. Available in API version 56.0 and later.
`enableCustomFlowsOnCycleCount`	boolean	Indicates whether Custom Flows on Cycle Count page for the Health Cloud app is enabled for your org (`true`) or not (`false`). Available in API version 56.0 and later.
`enableCustomFlowsOnExpiryPage`	boolean	Indicates whether Custom Flows on Expiry page for Health Cloud app is enabled for your org (`true`) or not (`false`). Available in API version 56.0 and later.
`enableDiseaseSurveillancePref`	boolean	Indicates whether the disease surveillance preference that monitors infectious diseases is enabled (`true`) or not (`false`). Available in API version 64.0 and later.
`enableHcCorePatientConsole`	boolean	Indicates whether to enable the OOTB Health Cloud Console App for Patients viewing.
`enableHcStdRelationshipJunctions`	boolean	Indicates whether to start using the core group membership settings instead of the ones coming from package for existing managed customers or for new ones to start using the relationships from core directly.
`enableHlsClinicalDcsnSuptAccessOrgPreference`	boolean	Indicates whether to enable Clinical Decision Support capabilities (OOTB IP, Data Model, and others.) (`true`) or not (`false`).

Field Name	Field Type	Description
enableHlsFhirSubscriptionSetting	boolean	Indicates whether to enable Documentation Template Rule (DTR) processes, data model underneath, and others for processing DTR use cases for Utilization Management & Clinical Decision Support users.
enableIndustriesLPIPreference	boolean	Indicates whether Industries Licensing, Permitting, and Inspections (LPI) feature is enabled for your org (`true`) or not (`false`). The default is `false`. This field is available in editions where Health Cloud is enabled. Available in API version 63.0 and later.
enableLifeSciencesClinialTrialManagement	boolean	Indicates whether to enable participant recruitment and enrollment to help organizations design and execute clinical trials and manage trial participant journeys.
enableMedicationManagementEnabled	boolean	Indicates whether Medication Management is enabled for your org (`true`) or not (`false`). The default is `false`.This field is available in API version 53.0 and later.
enableMedicalDeviceEnabled	boolean	Indicates whether the Intelligent Sales features are enabled (`true`) or not (`false`).
enableMedRecSetting	boolean	Indicates whether Medication Reconciliation is enabled for your org (`true`) or not (`false`). The default is `false`. This field is available in API version 54.0 and later.
enableMultipleCareProgramEnrolleeOrgPref	boolean	Indicates whether Multiple Care Program Enrollee is enabled for your org (`true`) or not (`false`). Available in API version 49.0 and later.
enableProviderSearchSyncOrgPref	boolean	Indicates whether provider data search is synced every six hours (`true`) or not (`false`). The default is `false`. This field is available in editions where Health Cloud is enabled.
enableRosterFileFeatureOrgPreference	boolean	Indicates whether this is the org pref used for the roster file in Provider Network Management (`true`) or not (`false`). The default is `false`.
enableTrialManagementConsentManagement	boolean	Indicates whether to enable consent management to help organizations take consent from prospective or enrolled candidates before storing their data.
enableUMPayerAppAccessOrgPreference	boolean	Indicates whether to enable the OOTB Console Utilization Management (UM) Payers App for UM Payers App Enablement. This setting can be enabled by Customers.
enableVisitInventoryEnabled	boolean	Indicates whether the visit data model is enabled (`true`) or not (`false`).
IsHomeHealthEnabled	boolean	Indicates whether users can schedule and execute visits for the home healthcare services they provide for their patients (`true`) or not (`false`).

Fields for Life Sciences Cloud

Field Name	Field Type	Description
`enableAdverseEvents`	boolean	Indicates whether Adverse Events and its data model is enabled for your org (`true`) or not (`false`). The default value is `false`. Available in API version 61.0 and later.
`enableComprehendMedical`	boolean	Indicates whether Comprehend Medical Analysis is enabled for your org (`true`) or not (`false`). The default value is `false`. This field is available in API version 54.0 and later.
`enableIESentimentAnalysis`	boolean	Indicates whether Industries Einstein Sentiment Insights is enabled for your org (`true`) or not (`false`). The default value is `false`. This field is available in API version 54.0 and later.
`enableLifeSciencesClinialTrailManagement`	boolean	Indicates whether Clinical Trial Management and its data model is enabled for your org (`true`) or not (`false`). Using this feature, organizations can design and execute clinical trials and manage trial participant journeys. The default value is `false`. Available in API version 61.0 and later.
`enableTrialManagementRandomization`	boolean	Indicates whether Research Study Randomization is enabled for your org (`true`) or not (`false`). Using this feature, users can design and run randomization procedures for their clinical trials. The default value is `false`. Available in API version 61.0 and later.

Fields for Automotive Cloud

Field Name	Field Type	Description
`enableCriteriaBasedSearchAndFilter`	boolean	Indicates whether Criteria-Based Search and Filter is enabled for your org. The default is `false`. This field is available in editions where Automotive Cloud is enabled.

Fields for Net Zero Cloud

Field Name	Description
`enableGnrcDisclsFrmwrk`	**Field Type** boolean **Description** Indicates whether the Industries Disclosure and Compliance Hub feature is enabled (`true`) or not (`false`) for your org. The default is `false`. Available in API version 57.0 and later in editions where Disclosure and Compliance Hub is enabled.

Field Name	Description
`enableInformationLibrary`	**Field Type** boolean **Description** Indicates whether the Information Library feature is enabled for your org. The default is `false`. Available in API version 59.0 and later in editions where Disclosure and Compliance Hub is enabled.
`enableMaterialityAssessment`	**Field Type** boolean **Description** Indicates whether the Manage Materiality Assessments feature is enabled for your org. The default is `false`. Available in API version 59.0 and later in editions where Net Zero Cloud is enabled.
`enableNZCMngEsgPgm`	**Field Type** boolean **Description** Indicates whether the Manage Environmental, Social, and Governance Programs feature is enabled for your org. The default is `false`. Available in API version 59.0 and later in editions where Net Zero Cloud is enabled.
`enableSCAssignFootprint`	**Field Type** boolean **Description** Indicates whether the Assign Carbon Footprint to Energy Use Records feature is enabled for your org. The default is `false`. Available in API version 54.0 and later in editions where Net Zero Cloud is enabled.
`enableSCBEIEnabled`	**Field Type** boolean **Description** Indicates whether the Manage Building Energy Intensity feature is enabled for your org. The default is `false`. Available in API version 54.0 and later in editions where Net Zero Cloud is enabled.
`enableSCCarbonAccounting`	**Field Type** boolean **Description** Indicates whether the Manage Carbon Accounting feature is enabled for your org. The default is `false`. Available in API version 54.0 and later in editions where Net Zero Cloud is enabled.

Field Name	Description
`enableSCCarbonCreditAlloc`	**Field Type** boolean **Description** Indicates whether the Allocate Carbon Credits to offset the unavoidable emissions feature is enabled for your org. The default is `false`. Available in API version 56.0 and later in editions where Net Zero Cloud is enabled.
`enableSCCreateFootprint`	**Field Type** boolean **Description** Indicates whether the Auto-Create Carbon Footprints feature is enabled for your org. The default is `false`. Available in API version 54.0 and later in editions where Net Zero Cloud is enabled.
`enableSCDGF`	**Field Type** boolean **Description** Indicates whether the Manage Data Gaps feature is enabled for your org. The default is `false`. Available in API version 54.0 and later in editions where Net Zero Cloud is enabled.
`enableSCExpansionUseCase`	**Field Type** boolean **Description** Indicates whether the Manage Carbon Accounting for Extended Organizational Boundaries feature is enabled for your org. The default is `false`. Available in API version 57.0 and later in editions where Net Zero Cloud is enabled.
`enableSCExternalEngMgmt`	**Field Type** boolean **Description** Indicates whether the Manage Supplier Sustainability Data feature is enabled for your org. The default is `false`. Available in API version 54.0 and later in editions where Net Zero Cloud is enabled.
`enableSCEmssnsForecasting`	**Field Type** boolean **Description** Indicates whether the Manage Carbon Emissions Forecast feature is enabled for your org. The default is `false`. Available in API version 54.0 and later in editions where Net Zero Cloud is enabled.

Field Name	Description
`enableSCSNGManagement`	**Field Type** boolean **Description** Indicates whether the Manage Social and Governance feature is enabled for your org. The default is `false`. Available in API version 57.0 and later in editions where Net Zero Cloud is enabled.
`enableSCScope3HubEnabled`	**Field Type** boolean **Description** Indicates whether the Manage Scope 3 Procurement Hub feature is enabled for your org. The default is `false`. Available in API version 54.0 and later in editions where Net Zero Cloud is enabled.
`enableSCTargetSetting`	**Field Type** boolean **Description** Indicates whether the Manage Emissions Target feature is enabled for your org. The default is `false`. Available in API version 54.0 and later in editions where Net Zero Cloud is enabled.
`enableSCWasteManagement`	**Field Type** boolean **Description** Indicates whether the Manage Waste-Related Data feature is enabled for your org. The default is `false`. Available in API version 54.0 and later in editions where Net Zero Cloud is enabled.
`enableSCWaterManagement`	**Field Type** boolean **Description** Indicates whether the Manage Water-Related Data feature is enabled for your org. For example, water consumption, withdrawal, and discharge. The default is `false`. Available in API version 56.0 and later in editions where Net Zero Cloud is enabled.
`enableSustainabilityCloud`	**Field Type** boolean **Description** Indicates whether the Net Zero Cloud feature is enabled for your org. The default is `false`. Available in API version 54.0 and later in editions where Net Zero Cloud is enabled.

Fields for Public Sector Solutions Features

Industries settings for Public Sector Solutions features are available in API version 57.0 and later.

Field Name	Field Type	Description
`enableBenefitAndGoalSharingPref`	boolean	Indicates whether the benefit and goal sharing feature is enabled for your org. The default is `false`. Available in editions where Public Sector Solutions is enabled.
`enableBenefitManagementPreference`	boolean	Indicates whether the program and benefit management feature is enabled for your org. The default is `false`. Available in editions where Public Sector Solutions is enabled.
`enableCarePlansPreference`	boolean	Indicates whether the care plan feature is enabled for your org. The care plan feature lets you create and edit care plans. The default is `false`. Available in editions where Public Sector Solutions is enabled. This field is available in API version 58.0 and later.
`enableEnhancedUIForISPref`		Indicates whether the enhanced interaction note interface feature is enabled for your org. The enhanced interaction note interface is a single guided flow that enables you to seamlessly create meeting notes; add interaction details, attendees, and interest tags; and share the notes with other users. The default is `false`. Available in editions where Public Sector Solutions is enabled. This field is available in API version 61.0 and later.

Fields for Salesforce Scheduler

Field Name	Field Type	Description
`appointmentDistributionOrgPref`	boolean	Indicates whether to schedule appointments for service resources based on appointment distribution (`true`) or not (`false`). The default value is `false`. Available in API version 52.0 and later.
`captureResourceUtilizationOrgPref`	boolean	Indicates whether to use a background process to calculate the usage of service resources from service appointments (`true`) or not (`false`). The default value is `false`. Available in API version 52.0 and later.
`enableAnyResourceTypeOrgPref`	boolean	Indicates whether to enable Salesforce Scheduler to consider service resource records with Agent resource type (`true`) or not (`false`). Before enabling this setting, create a service resource record as Main for each user, or update one of the service resource records as Main for each user. The default value is `false`. Available in API version 57.0 and later.
`enableAppFrmAnywhereOrgPref`	boolean	Indicates whether to use engagement channels for setting up shifts, work types, and booking a service appointment (`true`) or not (`false`). The default value is `false`. Available in API version 56.0 and later. See the prerequisites before you enable this setting.

Field Name	Field Type	Description
enableBlockResourceAvailabilityOrgPref	boolean	Indicates whether Salesforce Scheduler service appointments are added to users' Salesforce calendars. For example, if set to `false`, users don't see their service appointments on their calendars. The default is `false`. Available in API version 47.0 and later. This setting is used in Financial Services Cloud.
enableCapacitySchedulingPref	boolean	Indicates whether users can use capacity-based scheduling (`true`) or not (`false`). Use capacity-based scheduling to control the number of appointments that can be scheduled for a given shift and type of work. Available in API version 62.0 and later. See the prerequisite before you enable this setting.
enableCreateMultiAttendeeEventOrgPref	boolean	Indicates whether users can group individual events, and view the list of all attendees under a single event `true` or not `false`. The default is `false`. See the prerequisites before you enable this setting. Available in API version 55.0 and later. This setting is used in Financial Services Cloud.
enableDropInAppointmentsOrgPref	boolean	Indicates whether users can manage drop-in participants (`true`) or not (`false`). The default value is `false`. Available in API version 58.0 and later. See the prerequisite before you enable this setting.
enableDropInSkillMatchingOrgPref	boolean	Indicates whether skill and skill level matching is enabled for service resources that are assigned to waitlists for a service territory (`true`) or not (`false`). The default value is `false`. Available in API version 58.0 and later.
enableEventManagementOrgPref	boolean	Indicates whether users can add Salesforce Scheduler service appointments to their Salesforce calendars. The default is `false`. Available in API version 47.0 and later. This setting is used in Financial Services Cloud.
enableEventWriteOrgPref	boolean	Indicates whether to publish high-volume platform events when users create, update, or delete service appointments in Salesforce Scheduler (`true`) or not (`false`). If enabled, write these events to an external system to update it with Salesforce Scheduler service appointments. The default value is `false`. Available in API version 49.0 and later.
enableMultipleTopicsForShiftsOrgPref	boolean	Indicates whether the multiple topics for shifts feature is enabled (`true`) or disabled (`false`). The default value is `false`. Available in API version 56.0 and later. See the prerequisite before you enable this setting.
enableMultiResourceOrgPref	boolean	Indicates whether users can add multiple service resources to a service appointment. The default is `false` Available in API version 47.0 and later. This setting is used in Financial Services Cloud.

Field Name	Field Type	Description
enableOverbookingOrgPref	boolean	Indicates whether users can add multiple service appointments to a single time slot for a service resource. If set to false, concurrent time slots are visible, but can't be modified. The default is false Available in API version 47.0 and later. This setting is used in Financial Services Cloud.
enableShareSaWithArOrgPref	boolean	Indicates whether to share service appointments with assigned resources (true) or not (false). The default value is false. Available in API version 55.0 and later.
enableTopicOrTemplate	boolean	Indicates whether to use Salesforce Scheduler to manage Health Cloud appointments (true) or not (false). The default value is false. You must enable the enableTopicTimeSlot field before enabling this setting. Available in API version 52.0 and later.
enableTopicTimeSlot	boolean	Indicates whether to set operating hours for Service Territory Members for Work Type Groups (true) or not (false). The default value is false. Available in API version 52.0 and later. See the prerequisites before you enable this setting. After you enable this setting, you can't disable it.

Fields for Education Cloud

Field Name	Field Type	Description
enableEducationCloud	boolean	Indicates whether Education Cloud is enabled in Salesforce (true) or not (false). The default is false. Available in API version 57.0 and later in Developer, Enterprise, Performance and Unlimited editions.
enableStudentSuccess	boolean	Indicates whether Student Success is enabled in Salesforce (true) or not (false). The default is false. Available in API version 58.0 and later in Developer, Enterprise, Performance and Unlimited editions.
enableAcademicOperations	boolean	Indicates whether Academic Operations is enabled in Salesforce (true) or not (false). The default is false. Available in API version 59.0 and later in Developer, Enterprise, Performance and Unlimited editions.
enableAlumniRelations	boolean	Indicates whether Alumni Relations is enabled (true) or not (false). The default is false. Available in API version 59.0 and later in Developer, Enterprise, Performance and Unlimited editions.
enableMentoring	boolean	Indicates whether Mentoring is enabled (true) or not (false). The default is false. Available in API version 60.0 and later in Developer, Enterprise, Performance and Unlimited editions.

Declarative Metadata Sample Definition

The following is an example of an Industries.Settings metadata file.

```
<?xml version="1.0" encoding="UTF-8"?>
<IndustriesSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <enableCreateMultiAttendeeEventOrgPref>true</enableCreateMultiAttendeeEventOrgPref>
    <enableEventManagementOrgPref>true</enableEventManagementOrgPref>
    <enableMultiResourceOrgPref>false</enableMultiResourceOrgPref>
    <enableOverbookingOrgPref>true</enableOverbookingOrgPref>
    <enableBlockResourceAvailabilityOrgPref>true</enableBlockResourceAvailabilityOrgPref>

   <enableAccessToMasterListOfCoverageTypes>true</enableAccessToMasterListOfCoverageTypes>

    <enableManyToManyRelationships>true</enableManyToManyRelationships>

<allowMultipleProducersToWorkOnSamePolicy>false</allowMultipleProducersToWorkOnSamePolicy>

    <enableMortgageRlaTotalsOrgPref>true</enableMortgageRlaTotalsOrgPref>
    <enableObjectDetection>true</enableObjectDetection>
    <enableProviderSearchSyncOrgPref>true</enableProviderSearchSyncOrgPref>
    <enableReferralScoring>true</enableReferralScoring>
    <enableFSCInsuranceReport>true</enableFSCInsuranceReport>
    <enableSCCarbonAccounting>true</enableSCCarbonAccounting>
    <enableBenefitAndGoalSharingPref>true</enableBenefitAndGoalSharingPref>
    <enableBenefitManagementPreference>true</enableBenefitManagementPreference>
    <enableCarePlansPreference>true</enableCarePlansPreference>
</IndustriesSettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>Industries</members>
        <name>Settings</name>
    </types>
    <version>47.0</version>
</Package>
```

InterestTaggingSettings

Represents settings for Interest Tags, which your users can add to client records to capture client needs, interests, and prospecting opportunities.

Parent Type and Manifest Access

This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all the settings metadata types for the org are accessed using the "Settings" name. See Settings for more details.

File Suffix and Directory Location

`InterestTaggingSettings` values are stored in the `InterestTagging.settings` file in the `settings` folder. The `.settings` files are different from other named components, because there's only one settings file for each settings component.

Version

InterestTaggingSettings components are available in API version 54.0 and later.

Special Access Rules

Before you enable Interest Tags, you must enable Topics for Financial Services Cloud objects and assign Interest Tags permissions to users. See Interest Tags.

Fields

Field Name	Description
`enableInterestTagging`	**Field Type** boolean **Description** Enables Interest Tags in your org when set to `true`.

Declarative Metadata Sample Definition

The following is an example of an InterestTaggingSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<InterestTaggingSettings xmlns="http://soap.sforce.com/2006/04/metadata">
  <enableInterestTagging>true</enableInterestTagging>
</InterestTaggingSettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>InterestTagging</members>
        <name>Settings</name>
    </types>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

InventorySettings

Represents options for the Salesforce Omnichannel Inventory product.This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

InventorySettings values are stored in the `Inventory.settings` file in the `settings` directory. The `.settings` files are different from other named components because there is only one settings file for each settings component.

Version

Inventory settings are available in API version 51 and later.

Special Access Rules

This metadata type is only accessible by developers and customers using Salesforce Omnichannel Inventory.

Fields

Field Name	Field Type	Description
`enableOCIB2CIntegration`	boolean	Indicates whether Omnichannel Inventory is allowed to exchange inventory data with B2C Commerce (`true`) or not (`false`). The default value is `false`.
`enableOmniChannelInventory`	boolean	Indicates whether Omnichannel Inventory features are enabled (`true`) or not (`false`). The default value is `false`.

Declarative Metadata Sample Definition

The following is an example of an InventorySettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<InventorySettings xmlns="http://soap.sforce.com/2006/04/metadata"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <enableOmniChannelInventory>true</enableOmniChannelInventory>
    <enableOCIB2CIntegration>true</enableOCIB2CIntegration>
    </InventorySettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>Inventory</members>
        <name>Settings</name>
    </types>
```

```
    <version>51.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

InvLatePymntRiskCalcSettings

Represents the org's settings to identify the level of risks associated with payment of invoices.

Parent Type and Manifest Access

This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all the settings metadata types for the org are accessed using the "Settings" name. See Settings for more details.

File Suffix and Directory Location

`InvLatePymntRiskCalcSettings` values are stored in the `InvLatePymntRiskCalc.settings` file in the `settings` folder. The `.settings` files are different from other named components, because there's only one settings file for each settings component.

Version

InvLatePymntRiskCalcSettings components are available in API version 55.0 and later.

Fields

Field Name	Description
`enableInvLatePymntRiskCalc`	**Field Type** boolean **Description** Indicates the level of risk associated with payment of an invoice when the value is `true`.

Declarative Metadata Sample Definition

Theis example shows a sample InvLatePymntRiskCalcSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<InvLatePymntRiskCalcSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <enableInvLatePymntRiskCalc>true</enableInvLatePymntRiskCalc>
</InvLatePymntRiskCalcSettings>
```

This example shows a sample `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>InvLatePymntRiskCalc</members>
        <name>Settings</name>
    </types>
    <version>64.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

InvocableActionSettings

Represents the org's invocable action settings, such as whether partial save is allowed.This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

InvocableActionSettings values are stored in the `InvocableAction.settings` file in the settings directory. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

InvocableActionSettings components are available in API version 47.0 and later.

Fields

Field Name	Field Type	Description
`isPartialSaveAllowed`	boolean	Indicates whether partial save is enabled for most invocable actions that are invoked via REST API and executed in bulk. When the value is `true`, Salesforce tries three times to execute invocable actions that run successfully and rolls back only the invocable actions that fail to execute. This functionality is called partial save. If the field is set to `false`, if one invocable action fails, Salesforce rolls back other invocable actions in the same transaction and the entire transaction fails. Corresponds to the Enable Partial Save for Invocable Actions critical update.

Declarative Metadata Sample Definition

The following is an example of the InvocableAction.settings file.

```
<?xml version="1.0" encoding="UTF-8"?>
<InvocableActionSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <isPartialSaveAllowed>false</isPartialSaveAllowed>
</InvocableActionSettings>
```

Example Package Manifest

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
 <types>
  <members>InvocableAction</members>
  <name>Settings</name>
 </types>
 <version>47.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

KnowledgeSettings

Represents the metadata used to manage settings for Salesforce Knowledge.

This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for more details.

File Suffix and Directory Location

KnowledgeSettings values are stored in a single file named `Knowledge.settings` in the `settings` directory. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

KnowledgeSettings is available in API version 27.0 and later.

Fields

Field Name	Field Type	Description
`answers`	KnowledgeAnswerSettings	Represents the metadata used to manage settings for Salesforce Knowledge and Answers.
`cases`	KnowledgeCaseSettings	Represents the metadata used to manage settings for Salesforce Knowledge and Cases.
`defaultLanguage`	string	Required. The default language for Salesforce Knowledge. Use the abbreviation for the language, for example, en_US for United States English.
`enableChatterQuestionKBDeflection`	boolean	Indicates whether tracking for case deflection via Chatter is enabled (`true`) or not (`false`).
`enableCreateEditOnArticlesTab`	boolean	Indicates whether users can create and edit articles on the articles tab (`true`) or not (`false`).
`enableExternalMediaContent`	boolean	Indicates whether connecting to external media is enabled (`true`) or not (`false`).
`enableKbStandardSharing`	boolean	Indicates whether standard Salesforce sharing is enabled (`true`) or not (`false`).
`enableKnowledge`	boolean	Indicates whether Salesforce Knowledge is enabled (`true`) or not (`false`). This field is `false` by default.
`enableKnowledgeAgentContribution`	boolean	Indicates whether a user can create an article from a case (`true`) or not (`false`). (Classic only)
`enableKnowledgeArticleTextHighlights`	boolean	Indicates whether text snippet highlights in Salesforce Knowledge search results are enabled (`true`) or not (`false`). This field is `true` by default. Available in API version 47.0 and later.
`enableKnowledgeAnswersPromotion`	boolean	Indicates whether a user can create an article from a reply (`true`) or not (`false`). (Classic Only)
`enableKnowledgeCaseRL`	boolean	Indicates whether creating a list of cases linked to an article is enabled (`true`) or not (`false`). (Classic Only)
`enableKnowledgeKeywordAutoComplete`	boolean	Indicates whether auto-complete for keywords is enabled (`true`) or not (`false`) when searching Salesforce Knowledge. This field is `true` by default. Available in API version 47.0 and later.
`enableKnowledgeTitleAutoComplete`	boolean	Indicates whether auto-complete for article titles is enabled (`true`) or not (`false`) when

Field Name	Field Type	Description
		searching Salesforce Knowledge. This field is `true` by default. Available in API version 47.0 and later.
`enableLightningKbAutoLoadRichTextField`	boolean	Indicates whether rich text fields are enabled for editing when an article loads in Lightning Knowledge (`true`) or not (`false`). This field is `false` by default. Available in API version 47.0 and later.
`enableLightningKnowledge`	boolean	Indicates whether Lightning Knowledge is enabled (`true`) or not (`false`).
`languages`	KnowledgeLanguageSettings	A list of languages enabled for Salesforce Knowledge.
`showArticleSummariesCustomerPortal`	boolean	Indicates whether article summaries appear in the Customer Portal (`true`) or not (`false`).
`showArticleSummariesInternalApp`	boolean	Indicates whether article summaries appear in the internal knowledge base (`true`) or not (`false`).
`showArticleSummariesPartnerPortal`	boolean	Indicates whether article summaries appear in the partner portal (`true`) or not (`false`).
`showValidationStatusField`	boolean	Indicates whether validation status appears on articles (`true`) or not (`false`).
`suggestedArticles`	KnowledgeSuggestedArticlesSettings	Represents the metadata used to manage settings for the case fields used to suggest articles for cases. Available in API version 37.0 and later.
`votingEnabled`	boolean	When `true`, enables users to vote for a product or feature that uses Vote, such as Articles in Knowledge. Available in API version 50.0 and later.

KnowledgeAnswerSettings

Represents the metadata used to manage settings for Salesforce Knowledge and Answers.

Field Name	Field Type	Description
`assignTo`	string	Specifies the username an article is assigned to from Answers.
`defaultArticleType`	string	The default article type for articles created from Answers. Uses the API name of the article type.
`enableArticleCreation`	boolean	Indicates whether users can create articles from Answers (`true`) or not (`false`).

KnowledgeCaseSettings

Represents the metadata used to manage settings for Salesforce Knowledge and Cases.

Field Name	Field Type	Description
`articlePDFCreationProfile`	string	The profile used to create a PDF of an article from Cases.
`articlePublicSharingSites`	KnowledgeSitesSettings	Represents the metadata used to manage settings for Salesforce Knowledge and Sites.
`articlePublicSharingCommunities`	KnowledgeSitesSettings	Represents the metadata used to manage settings for Salesforce Knowledge and Experience Cloud sites.
`articlePublicSharingSitesChatterAnswers`	KnowledgeSitesSettings	Represents the metadata used to manage settings for Salesforce Knowledge and Sites with Chatter Answers.
`assignTo`	string	Specifies the username an article is assigned to from Cases.
`customizationClass`	string	Specifies the Apex class used for customization.
`defaultContributionArticleType`	string	The default article type for articles created from Cases.
`editor`	KnowledgeCaseEditor (enumeration of type string)	Indicates the rich text editor type. Valid values are: • `simple` • `standard`
`enableArticleCreation`	boolean	Indicates whether users can create articles from Cases (`true`) or not (`false`). Controls whether other fields on KnowledgeCaseSettings can be set.
`enableArticlePublicSharingSites`	boolean	Indicates whether articles can be shared via a public site (URL) from Cases (`true`) or not (`false`).
`enableCaseDataCategoryMapping`	boolean	Indicates whether Case Data Category mapping is enabled (`true`) or not (`false`).
`useProfileForPDFCreation`	boolean	Indicates whether a profile is used to create a PDF of an article from Cases (`true`) or not (`false`).

KnowledgeSitesSettings

Represents the metadata used to manage settings for Salesforce Knowledge and Sites.

Field Name	Field Type	Description
`site`	string[]	Specifies the site used for Salesforce Knowledge and Sites.

KnowledgeLanguageSettings

A list of languages enabled for Salesforce Knowledge. KnowledgeLanguageSettings is available in API version 28.0 and later.

Field Name	Field Type	Description
`language`	KnowledgeLanguage[]	Represents the metadata used to manage settings for the languages enabled for Salesforce Knowledge.

KnowledgeLanguage

Represents the metadata used to manage settings for the languages enabled for Salesforce Knowledge. KnowledgeLanguage is available in API version 28.0 and later.

Field Name	Field Type	Description
`active`	boolean	Indicates whether the language is enabled (`true`) or not (`false`).
`defaultAssignee`	string	The default assignee for articles in the language.
`defaultAssigneeType`	KnowledgeLanguageLookupValueType (enumeration of type string)	Indicates the default assignee type. Valid values are: • `User` • `Queue`
`defaultReviewer`	string	The default reviewer for articles in the language.
`defaultReviewerType`	KnowledgeLanguageLookupValueType (enumeration of type string)	Indicates the default reviewer type. Valid values are: • `User` • `Queue`
`name`	string	The code for the language name, for example: English is `en`. See "What languages does Salesforce support?" in the Salesforce Help for a list of supported languages and their codes.

KnowledgeSuggestedArticlesSettings

Represents the metadata used to manage settings for the articles suggested for cases, work orders, and work order line items. The Work Order and Work Order Line Item objects must be enabled in the org to use the associated fields.

Field Name	Field Type	Description
`caseFields`	KnowledgeCaseFieldsSettings	Represents a list of the case fields used to suggest articles for the case.
`useSuggestedArticlesForCase`	boolean	Indicates whether case content is used to suggest articles for cases (true) or not (false).

Field Name	Field Type	Description
`workOrderFields`	KnowledgeWorkOrderFieldsSettings	Represents a list of the work order fields used to suggest articles for the work order.
`workOrderLineItemFields`	KnowledgeWorkOrderLineItemFieldsSettings	Represents a list of the work order line item fields used to suggest articles for the work order line item.

KnowledgeCaseFieldsSettings

Represents a list of the case fields used to suggest articles for the case. Available in API version 37.0 and later.

Field Name	Field Type	Description
`field`	KnowledgeCaseField[]	Specifies the names of the case fields used to suggest articles for the case.

KnowledgeCaseField

Represents the name of the case field used to suggest articles for the case. Available in API version 37.0 and later.

Field Name	Field Type	Description
`name`	string	Specifies the name of the case field used to suggest articles for the case.

KnowledgeWorkOrderFieldsSettings

Represents a list of the work order fields used to suggest articles for the work order. Available in API version 39.0 and later.

Field Name	Field Type	Description
`field`	KnowledgeWorkOrderField[]	Specifies the names of the work order fields used to suggest articles for the work order.

KnowledgeWorkOrderField

Represents the name of the work order field used to suggest articles for the work order. Available in API version 39.0 and later.

Field Name	Field Type	Description
`name`	string	Specifies the name of the work order field used to suggest articles for the work order.

KnowledgeWorkOrderLineItemFieldsSettings

Represents a list of the work order line item fields used to suggest articles for the work order line item. Available in API version 39.0 and later.

Field Name	Field Type	Description
field	KnowledgeWorkOrderLineItemField[]	Specifies the names of the work order line item fields used to suggest articles for the work order line item.

KnowledgeWorkOrderLineItemField

Represents the name of the work order line item field used to suggest articles for the work order line item. Available in API version 39.0 and later.

Field Name	Field Type	Description
name	string	Specifies the name of the work order line item field used to suggest articles for the work order line item.

Declarative Metadata Sample Definition

This is a sample Knowledge settings file.

```
<?xml version="1.0" encoding="UTF-8"?>
<KnowledgeSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <answers>
        <enableArticleCreation>false</enableArticleCreation>
    </answers>
    <cases>
        <articlePDFCreationProfile>partner portal knowledge
profile</articlePDFCreationProfile>
        <articlePublicSharingSites>
            <site>KnowledgeSite</site>
            <site>PKB2Site</site>
            <site>ChatterAnswersSite</site>
        </articlePublicSharingSites>
        <articlePublicSharingSitesChatterAnswers>
            <site>ChatterAnswersSite</site>
        </articlePublicSharingSitesChatterAnswers>
        <assignTo>testall@kb.org</assignTo>
        <defaultContributionArticleType>Support</defaultContributionArticleType>
        <editor>simple</editor>
        <enableArticleCreation>true</enableArticleCreation>
        <enableArticlePublicSharingSites>true</enableArticlePublicSharingSites>
        <useProfileForPDFCreation>true</useProfileForPDFCreation>
    </cases>
    <defaultLanguage>ja</defaultLanguage>
    <enableCreateEditOnArticlesTab>true</enableCreateEditOnArticlesTab>
    <enableExternalMediaContent>true</enableExternalMediaContent>
    <enableKnowledge>true</enableKnowledge>
    <showArticleSummariesCustomerPortal>true</showArticleSummariesCustomerPortal>
    <showArticleSummariesInternalApp>true</showArticleSummariesInternalApp>
    <showArticleSummariesPartnerPortal>true</showArticleSummariesPartnerPortal>
    <showValidationStatusField>true</showValidationStatusField>
    <suggestedArticles>
      <caseFields>
```

```
        <field>
           <name>Subject</name>
        </field>
        <field>
           <name>SuppliedEmail</name>
        </field>
      </caseFields>
      <useSuggestedArticlesForCase>true</useSuggestedArticlesForCase>
    </suggestedArticles>
</KnowledgeSettings>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

LanguageSettings

Represents an organization's language settings. Language settings control end-user language selection, locale formats, and translation options. This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

LanguageSettings values are stored in the `Language.settings` file in the `settings` directory. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

LanguageSettings is available in API version 47.0 and later.

Fields

Field Name	Field Type	Description
`enableCanadaIcuFormat`	boolean	Indicates whether the ICU format is enabled for the en_CA locale (`true`) or not (`false`). This field has a default value of `true` for orgs created in API version 47.0 and later. Orgs created prior to API version 47.0 have a default of `false`.
`enableDataTranslation`	boolean	Indicates whether data translation is enabled (`true`) or not (`false`). This field has a default value of `false`. This field is available in API version 49.0 and later.
`enableEndUserLanguages`	boolean	Indicates whether end-user languages are enabled (`true`) or not (`false`). This field has a default value of `false`.

Field Name	Field Type	Description
enableICULocaleDateFormat	boolean	Indicates whether date and currency are formatted with the International Components for Unicode (true) or not (false). This field has a default value of true for orgs created in API version 47.0 and later. Orgs created prior to API version 47.0 have a default of false. See Go Global with New International Locale Formats for more information.
enableLanguageExtensionPackage (beta)	boolean	Indicates whether the org has enabled language extension packages (true) or not (false). Language extension packages contain translations of components in other packages. This field has a default value of false. This field is available in API version 58.0 and later. Note: enableLanguageExtensionPackage (beta) is a pilot or beta service that is subject to the Beta Services Terms at Agreements - Salesforce.com or a written Unified Pilot Agreement if executed by Customer, and applicable terms in the Product Terms Directory. Use of this pilot or beta service is at the Customer's sole discretion.
enableLocaleInsensitiveFiltering	boolean	Indicates whether users can filter query results, regardless of the locale or language associated with the user (true) or if they can't filter results (false). This field has a default value of false. This field is available in API version 56.0 and later.
enableLocalNamesForStdObjects	boolean	Indicates whether local name fields can be defined for standard objects (true) or not (false). This field has a default value of false. This field is available in API version 48.0 and later.
enablePlatformLanguages	boolean	Indicates whether platform-only languages are enabled (true) or not (false). This field has a default value of false. Setting this field to true also sets enableEndUserLanguages to true.
enableTranslationWorkbench	boolean	Indicates whether the Translation Workbench is enabled (true) or not (false). This field has a default value of false.
useLanguageFallback	boolean	Indicates whether translation follows the language fallback rule (true) or returns the primary label (false). This field has a default value of true.

Declarative Metadata Sample Definition

The following is an example of a LanguageSettings file.

```
<?xml version="1.0" encoding="UTF-8"?>
<LanguageSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <enableCanadaIcuFormat>true</enableCanadaIcuFormat>
    <enableDataTranslation>false</enableDataTranslation>
    <enableEndUserLanguages>true</enableEndUserLanguages>
    <enableICULocaleDateFormat>true</enableICULocaleDateFormat>
    <enableLocalNamesForStdObjects>false</enableLocalNamesForStdObjects>
```

```
    <enablePlatformLanguages>false</enablePlatformLanguages>
    <enableTranslationWorkbench>true</enableTranslationWorkbench>
    <useLanguageFallback>true</useLanguageFallback>
</LanguageSettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>Language</members>
        <name>Settings</name>
    </types>
    <version>49.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

LeadConfigSettings

Represents configuration settings for Leads that control how they are converted and displayed, and what actions are available. This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

LeadConfigSettings values are stored in the `LeadConfig.settings` file in the `settings` folder. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

LeadConfigSettings is available in API versions 47.0 and later.

Fields

Field Name	Field Type	Description
doesEnableLeadConvertDefaultSubjectBlankTaskCreation	boolean	Configures whether tasks without a subject are created during lead conversion. If `true`, tasks are created when the default subject field has no value. If `false`, only tasks with a subject are created.
doesHideOpportunityInConvertLeadWindow	boolean	Hides the opportunity section of the Convert Lead window during the conversion of a lead. Default value is `false`.

Field Name	Field Type	Description
doesPreserveLeadStatus	boolean	If your organization uses record types, the lead status changes to the lead status value of the new owner's record type during conversion. Set doesPreserveLeadStatus to true to preserve the value of the lead status during conversion. Orgs that use record types can create a lead process that allows different lead status values for different record types. If doesPreserveLeadStatus is false, the lead status might change during lead conversion if the new owner's record type has a different default value for lead status. Default value is true.
doesSelectNoOpportunityOnConvertLead	boolean	Prevents an opportunity from being created when the lead is converted. Default value is false.
doesTrackHistory	boolean	Enables field history tracking for leads. When field history tracking is enabled, users can choose the fields they want to track. Default value is false.
enableConversionsOnMobile	boolean	Lets a user convert leads on their mobile devices. The Convert Lead action converts qualified leads to contacts. Default value is true.
enableOrgWideMergeAndDelete	boolean	Lets a user merge and delete leads. The user must also have the Public Read/Write/Transfer permission. Default value is false.
shouldLeadConvertRequireValidation	boolean	Enforces validation rules when converting leads. Default value is true.
shouldSendNotificationEmailWhenLeadOwnerUpdatedViaApexInLEX	boolean	Indicates whether an email notification is sent when the lead owner is updated through Apex in Lightning Experience. Default value is false.

Declarative Metadata Sample Definition

The following is an example of the LeadConfigSettings type:

```
<?xml version="1.0" encoding="UTF-8"?>
<LeadConfigSettings xmlns="http://soap.sforce.com/2006/04/metadata">
<doesEnableLeadConvertDefaultSubjectBlankTaskCreation>false</doesEnableLeadConvertDefaultSubjectBlankTaskCreation>
<doesHideOpportunityInConvertLeadWindow>false</doesHideOpportunityInConvertLeadWindow>
<doesPreserveLeadStatus>true</doesPreserveLeadStatus>
<doesSelectNoOpportunityOnConvertLead>false</doesSelectNoOpportunityOnConvertLead>
<doesTrackHistory>false</doesTrackHistory>
<enableConversionsOnMobile>true</enableConversionsOnMobile>
```

```
<enableOrgWideMergeAndDelete>false</enableOrgWideMergeAndDelete>
<shouldLeadConvertRequireValidation>true</shouldLeadConvertRequireValidation>
<shouldSendNotificationEmailWhenLeadOwnerUpdatesViaApexInLEX>false</shouldSendNotificationEmailWhenLeadOwnerUpdatesViaApexInLEX>
</LeadConfigSettings>
```

Example Package Manifest

The following is an example package manifest used to deploy or retrieve the LeadConfig settings metadata for an organization:

```
<?xml version="1.0" encoding="UTF-8"?>
<!--

<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>LeadConfig</members>
        <name>Settings</name>
    </types>
    <version>61.0</version>
</Package>
```

LeadConvertSettings

Represents an organization's custom field mappings for lead conversion. Custom fields can be mapped from Leads to Accounts, Contacts, and Opportunities. Options for creating opportunities during lead conversion can also be specified. This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

LeadConvertSettings components have the suffix `LeadConvertSetting` and are stored in the `LeadConvertSettings` folder.

Version

LeadConvertSettings is available in API versions 39.0 and later.

Fields

Field Name	Description
`allowOwnerChange`	**Field Type** boolean **Description** Indicates whether to include the **Record Owner** field in the **Convert Lead** dialog box (`true`) or not (`false`).
`objectMapping`	**Field Type** ObjectMapping[]

Field Name	Description
	Description A set of custom field mappings between objects. Up to three `objectMapping` types can be declared—one each for account, contact, and opportunity.
`opportunityCreationOptions`	**Field Type** VisibleOrRequired (enumeration of type string) **Description** This field determines whether the **Opportunity** field is visible or required in the **Convert Lead** dialog box. Values are: • `VisibleOptional`—The **Opportunity** field is included in the dialog box but not required. A new opportunity is created if the user enters an opportunity name. This is the default value. • `VisibleRequired`—The **Opportunity** field is included in the dialog box and is required. A new opportunity is created based on the name the user enters. • `NotVisible`—The **Opportunity** field is not included in the dialog box. No opportunity is created.

ObjectMapping

Represents a custom field mapping between two objects.

Field Name	Description
`inputObject`	**Field Type** string **Description** Required. The name of the object type containing the source fields for mapping. The value is always `Lead`.
`mappingFields`	**Field Type** ObjectMappingField[] **Description** A set of input and output field names of the custom fields to be mapped.
`outputObject`	**Field Type** string **Description** Required. The object type receiving data during lead conversion.

Field Name	Description
	• `Account` • `Contact` • `Opportunity`

ObjectMappingField

Represents custom field names to be mapped between objects.

Field Name	Description
`inputField`	**Field Type** string **Description** Required. The name of a custom lead field supplying source data during lead conversion.
`outputField`	**Field Type** string **Description** Required. The name of a custom account, contact, or opportunity field that will receive data from source field named in the accompanying `inputField` entry.

Declarative Metadata Sample Definition

The following is an example of the LeadConvertSettings type:

```
<?xml version="1.0" encoding="UTF-8"?>
<LeadConvertSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <allowOwnerChange>false</allowOwnerChange>
    <objectMapping>
        <inputObject>Lead</inputObject>
        <mappingFields>
            <inputField>custom_lead_field_1</inputField>
            <outputField>custom_account_field_1</outputField>
        </mappingFields>
        <mappingFields>
            <inputField>custom_lead_field_2</inputField>
            <outputField>custom_account_field_2</outputField>
        </mappingFields>
        <mappingFields>
            <inputField>custom_lead_field_3</inputField>
            <outputField>custom_account_field_3</outputField>
        </mappingFields>
```

```
        <outputObject>Account</outputObject>
    </objectMapping>
    <objectMapping>
        <inputObject>Lead</inputObject>
        <mappingFields>
            <inputField>custom_lead_field_4</inputField>
            <outputField>custom_opportunity_field_1</outputField>
        </mappingFields>
        <outputObject>Opportunity</outputObject>
    </objectMapping>
    <opportunityCreationOptions>VisibleOptional</opportunityCreationOptions>
</LeadConvertSettings>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

LiveAgentSettings

Represents an organization's Chat settings, such as whether Chat is enabled. This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

LiveAgentSettings values are stored in the `LiveAgent.settings` file in the `settings` directory. The `.settings` files are different from other named components because there's only one settings file for each settings component.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

Version

LiveAgentSettings is available in API version 28.0 and later.

Fields

Field Name	Field Type	Description
`enableChatFindOrCreateEnable`	boolean	Indicates whether the `findOrCreate` method of the Chat API is enabled for agents (`true`) or not (`false`). Available in API version 53.0 and later.
`enableLiveAgent`	boolean	Indicates whether Chat is enabled (`true`) or not (`false`).
`enableQuickTextEnabled`	boolean	Indicates whether Quick Text is enabled (`true`) or not (`false`).
`priority`	integer	Indicates the priority level of a Chat.

Declarative Metadata Sample Definition

This is a sample Chat settings file.

```
<?xml version="1.0" encoding="UTF-8"?>
<LiveAgentSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <enableLiveAgent>true</enableLiveAgent>
</LiveAgentSettings>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SEE ALSO:

Service Cloud Chat Developer Guide: findOrCreate

LightningExperienceSettings

Represents the settings that modify an org's Lightning Experience configuration. This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

A LightningExperienceSettings component has the suffix `.settings` and is stored in the `settings` folder. The `.settings` files are different from other named components because there is only one settings file for each settings component.

Version

LightningExperienceSettings components are available in API version 47.0 and later.

Fields

Field Name	Field Type	Description
`activeThemeName`	string	If specified, indicates the API name of the custom LightningExperienceTheme that's currently active in the org. If the field isn't specified, then the org uses one of the built-in themes that Salesforce provides. Custom themes are available in API version 48.0 and later.
`enableAccessCheckCrucPref`	boolean	Deprecated in API version 47.0 and later because the feature is no longer available.
`enableApiUserLtngOutAccessPref`	boolean	Deprecated in API version 48.0 and later because the feature is no longer available. This field corresponds to the API Only Users Can Access Only Salesforce APIs critical update, which was enforced in Spring '20. If a user has the API Only User permission, they can access Salesforce only via

Field Name	Field Type	Description
		APIs, regardless of their other permissions. This restriction already applied to other Salesforce features, but the critical update enforced the restriction in Lightning Out also.
`enableAuraCDNPref`	boolean	Indicates whether Lightning Experience and other apps use a content delivery network (CDN) to serve the static content for Lightning Component framework. A CDN generally speeds up page load time, but it also changes the source domain that serves the files. If your company has IP range restrictions for content served from Salesforce, test thoroughly before enabling this setting. The default is `true`.
`enableAuraDepAccessChksCRUCPref`	boolean	Removed in API version 51.0 and later because the feature is no longer available.
`enableAuraSecStaticResCRUCPref`	boolean	Indicates whether the Enable Secure Static Resources for Lightning Components release update is enforced (`true`) or not (`false`). To improve security, this update serves all static resources from the `visualforce` domain instead of the `lightning` domain. This change prevents a script included in a static resource from accessing the document in the `lightning` domain due to the same-origin security policy. This field is available in API version 50.0 and later.
`enableDeferRenderingWorkspacePage`	boolean	Reserved for future use.
`enableErrorExperienceEnabled`	boolean	Reserved for future use.
`enableFeedbackInMobile`	boolean	Indicates whether users can send feedback to Salesforce from the mobile app. The default is `false`.
`enableGoogleSheetsForSfdcEnabled`	boolean	Reserved. Do not use.
`enableIE11DeprecationMsgHidden`	boolean	Deprecated in API version 47.0 and later because the feature is no longer available.
`enableIE11LEXCrucPref`	boolean	Deprecated in API version 47.0 and later because the feature is no longer available.
`enableInAppLearning`	boolean	Indicates whether Learning Paths is enabled (`true`) or not (`false`). The default is `true`. If `false`, you can't assign learning items to users, and users can't view assigned learning items in the Guidance Center or access Learning Home. This field is available in API version 54.0 and later. See Learning Paths in Salesforce Help.
`enableInAppTooltips`	boolean	Indicates whether users see onboarding tips in the mobile app. The default is `false`.
`enableLEXOniPadEnabled`	boolean	Indicates whether Lightning Experience is turned on for iPad Browsers (`true`) or not (`false`). The default is `false`. See Give Users Access to Lightning Experience on iPad Browsers (Beta) in Salesforce Help.
`enableLexEndUsersNoSwitching`	boolean	Indicates whether Salesforce Classic is turned off for your org (`true`) or not (`false`). Removes the Switcher for all users in the org. The default is `false`. See Turn Off Salesforce Classic for Your Org in Salesforce Help.

Field Name	Field Type	Description
		This field is similar to `enableUsersAreLightningOnly`. If either field is set to `true`, users are blocked from switching to Salesforce Classic.
`enableLightningPreviewPref`	boolean	Indicates whether Local Dev is turned on for your org (`true`) or not (`false`). The default value is `false`. See Preview Components with Local Dev in Salesforce Developers. This field is available in API version 62.0 and later.
`enableNavPersonalizationOptOut`	boolean	Indicates whether users are blocked from personalizing the Lightning Experience navigation bar (`true`) or not (`false`). The default is `false` (that is, users can personalize the navigation bar by default). Salesforce recommends disabling personalization at the app level, not the org level. See Configure User Interface Settings in Salesforce Help.
`enableNoBackgroundNavigations`	boolean	Indicates whether consecutive API navigation calls in Visualforce pages are allowed (`false`) or blocked (`true`). The default is `false`.
`enableQuip`	boolean	Indicates whether Quip is available for your org. This field is available in API version 51.0 and later.
`enableRemoveThemeBrandBanner`	boolean	Deprecated in API version 47.0 and later because the feature is no longer available.
`enableS1BannerPref`	boolean	Deprecated in API version 47.0 and later because the feature is no longer available.
`enableS1BrowserEnabled`	boolean	Indicates whether all users can access the Salesforce mobile web view from a supported mobile browser (`true`) or not (`false`). If `false`, then users must access the Salesforce mobile full site view from a mobile browser. Full site view doesn't have the full functionality of mobile web view. Salesforce Classic and Lightning Experience aren't supported on mobile browsers.
`enableS1DesktopEnabled`	boolean	Indicates whether Lightning Experience is turned on in the org (`true`) or not (`false`). After it's enabled, this setting can't be disabled via the user interface or the API. See Turn on Lightning Experience for Your Org in Salesforce Help.
`enableS1UiLoggingEnabled`	boolean	Deprecated in API version 47.0 and later because the feature is no longer available.
`enableSidToken3rdPartyAuraApp`	boolean	Reserved for internal use.
`enableSkypeChatEnabled`	boolean	Deprecated in API version 50.0 and later because the feature is no longer available.
`enableSparkAllUsers`	boolean	Deprecated in API version 50.0 and later because the feature is no longer available.
`enableSparkConversationEnabled`	boolean	Deprecated in API version 50.0 and later because the feature is no longer available.

Field Name	Field Type	Description
enableSplitViewOnStandard	boolean	Removed in API version 52.0 and later because the feature is no longer available.
enableStackedModalManagerEnabled	boolean	Indicates whether the Enable LWC Stacked Modals release update is enabled (`true`) or not (`false`). For orgs created before Summer '24, the default value is `false`. For orgs created in Summer '24 and later, the default value is `true`. This field is available in API version 61.0 and later.
enableTryLightningOptOut	boolean	Indicates whether the Try Lightning Experience Now prompt is hidden from users (`true`) or not (`false`). The default is `false`. See Try Lightning Experience Now Prompt in Salesforce Help.
enableUseS1AlohaDesktop	boolean	Deprecated in API version 47.0 and later because the feature is no longer available.
enableUsersAreLightningOnly	boolean	Indicates whether Salesforce Classic is turned off for your org (`true`) or not (`false`). Removes the Switcher for all users in the org. The default is `false`. This field is similar to `enableLexEndUsersNoSwitching`. If either field is set to `true`, users are blocked from switching to Salesforce Classic.
enableWebExEnabled	boolean	Deprecated in API version 50.0 and later because the feature is no longer available.
enableWebexAllUsers	boolean	Deprecated in API version 50.0 and later because the feature is no longer available.
isLEXExtensionComponentCustomizationOff	boolean	Indicates whether all users can enable the Lightning Extension Component Customization feature. If `false`, the feature is disabled for all users, even users who had it enabled. See Try New Features with the Lightning Extension for Chrome. This field is available in API version 48.0 and later.
isLEXExtensionDarkModeOff	boolean	Indicates whether all users can enable the Lightning Extension Dark Mode feature. If `false`, the feature is disabled for all users, even users who had it enabled. This field is available in API version 48.0 and later.
isLEXExtensionLinkGrabberOff	boolean	Indicates whether all users can enable the Lightning Extension Link Grabber feature. If `false`, the feature is disabled for all users, even users who had it enabled. This field is available in API version 48.0 and later.
isLEXExtensionOff	boolean	Indicates whether all users can enable the Lightning Extension for your org. If `false`, your users can't enable the Lightning Extension, even if they already have it installed. This field is available in API version 48.0 and later.

Declarative Metadata Sample Definition

The following is an example of a LightningExperienceSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<LightningExperienceSettings xmlns="http://soap.sforce.com/2006/04/metadata">

    <enableAuraCDNPref>true</enableAuraCDNPref>
    <enableFeedbackInMobile>true</enableFeedbackInMobile>
    <enableInAppTooltips>true</enableInAppTooltips>
    <enableLEXOnIpadEnabled>true</enableLEXOnIpadEnabled>
    <enableLexEndUsersNoSwitching>true</enableLexEndUsersNoSwitching>
    <enableNavPersonalizationOptOut>true</enableNavPersonalizationOptOut>
    <enableS1BrowserEnabled>false</enableS1BrowserEnabled>
    <enableS1DesktopEnabled>true</enableS1DesktopEnabled>
    <enableTryLightningOptOut>true</enableTryLightningOptOut>
    <enableUsersAreLightningOnly>true</enableUsersAreLightningOnly>
</LightningExperienceSettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>LightningExperience</members>
        <name>Settings</name>
    </types>
    <version>47.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

LiveMessageSettings

Represents an org's LiveMessage settings.

Version

LiveMessageSettings components are available in API version 42.0 and later.

File Suffix and Directory Location

LiveMessageSettings values are stored in the LiveMessage.settings file in the settings folder. The .settings files are different from other named components because there is only one settings file for each settings component.

Fields

Field Name	Field Type	Description
`enableCheckCEUserPerm`	boolean	Gives access to ConversationEntry objects only to users with the Access Conversation Entries user permission enabled (true) or to all users (false) in an org. For orgs created before API version 50.0, the default value is `false`. For orgs created on or after API version 50.0, the default value is `true`.
`enableLiveMessage`	boolean	Turns LiveMessage on (`true`) or off (`false`) in an org. The default value is `false`.

Declarative Metadata Sample Definition

The following is an example of a `liveMessageSettings` component.

```
<?xml version="1.0" encoding="UTF-8"?>
<LiveMessageSettings xmlns="http://soap.sforce.com/2006/04/metadata">
   <enableLiveMessage>true</enableLiveMessage>
</LiveMessageSettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>LiveMessage</members>
        <name>Settings</name>
    </types>
    <version>44.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

MacroSettings

Represents an organization's Macro settings, such as whether or not folders is enabled. This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

MacroSettings values are stored in the `Macro.settings` file in the `settings` directory. The `.settings` files are different from other named components because there's only one settings file for each settings component.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

Version

MacroSettings is available in API version 39.0 and later.

Fields

Field Name	Field Type	Description
`enableAdvancedSearch`	boolean	Indicates whether users can search all macro text fields (`true`) or not (`false`).
`macrosInFolders`	boolean	Indicates whether users can organize and share macros using folders (`true`) or not (`false`). Available in API version 44.0 and later.

Declarative Metadata Sample Definition

This is a sample Macro settings file.

```
<?xml version="1.0" encoding="UTF-8"?>
<MacroSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <enableAdvancedSearch>true</enableAdvancedSearch>
    <macrosInFolders>true</macrosInFolders>
</MacroSettings>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

MailMergeSettings

Represents the settings for Extended Mail Merge functionality.

File Suffix and Directory Location

A MailMerge component file has the suffix `MailMerge.settings` and is stored in the `settings` directory.

Version

MailMergeSettings components are available in API version 51.0 and later. Before API version 51.0, fields from MailMergeSettings were found within OrgSettings components.

Fields

Field Name	Field Type	Descriptions

Field Name	Field Type	Descriptions
enableExtendedMailMerge	boolean	Indicates whether the Salesforce Classic product, Extended Mail Merge, is enabled (true) or not (false). Use Extended Mail Merge to generate Microsoft Word documents — such as form letters or address labels — from Salesforce records using Word document templates. Default value is false.
saveMailMergeDocsAsSalesforceDocs	boolean	Indicates whether mail-merged documents are saved to the My Personal Documents folder of the user who generated the mail merge. (true) or not (false). If (false), only documents over 3 MB are saved to the user's documents folder. Smaller documents are emailed to the user. Default value is (false).

Declarative Metadata Sample Definition

The following is an example of a MailMergeSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<MailMergeSettings xmlns="http://soap.sforce.com/2006/04/metadata"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <enableExtendedMailMerge>true</enableExtendedMailMerge>
</MailMergeSettings>
```

Example Package Manifest

The following is an example package manifest used to deploy or retrieve mail merge settings metadata for an organization:

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <name>MailMergeSettings</name>
    </types>
    <version>51.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character * (asterisk) in the package.xml manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

MapAndLocationSettings

Represents an org's map and location settings.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

Declarative Metadata File Suffix and Directory Location

MapAndLocationSettings values are stored in a single file named `Mapandlocation.settings` in the `settings` directory. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

Map and location settings are available in API version 46.0 and later.

Fields

Field	Field Type	Description
`enableAddressAutoComplete`	boolean	Indicates whether auto-complete is enabled on address fields (`true`) or not (`false`).
`enableMapsAndLocation`	boolean	Indicates whether the maps and location services are enabled (`true`) or not (`false`)

Declarative Metadata Sample Definition

This is a sample `mapandlocation.settings` metadata file.

```
<?xml version="1.0" encoding="UTF-8"?>
<MapsAndLocationSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <enableAddressAutoComplete>false</enableAddressAutoComplete>
    <enableMapsAndLocation>false</enableMapsAndLocation>
</MapsAndLocationSettings>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

MeetingsSettings

Represents the settings to enable Salesforce Meetings and the integration with Zoom video conferencing.

Version

MeetingsSettings components are available in API version 51.0 and later.

Special Access Rules

The MeetingsSettings type isn't available in scratch orgs.

Fields

Field Name	Description
`enableSalesforceMeetings`	**Field Type** boolean **Description** Indicates whether the Salesforce Meetings feature is enabled (`true`) or not (`false`). When set to `true`, Salesforce admins can assign the Salesforce Meetings user permission to grant users access to the Meeting Digest and other Salesforce Meetings features. The default value is `false`.
`enableSalesforceMeetingsSyncCheck`	**Field Type** boolean **Description** Indicates whether your company uses an activity sync solution (`true`) or hasn't made that indication (`false`). Indicating your company uses an activity solution such as Einstein Activity Capture is required to enable Salesforce Meetings. The default value is `false`.
`enableZoomVideoConference`	**Field Type** boolean **Description** Indicates whether users can connect their company Zoom accounts to Salesforce (`true`) or not (`false`). When set to `true`, Zoom can be added as a recording in Einstein Conversation Insights. The default value is `false`.

Declarative Metadata Sample Definition

The following is an example of a MeetingsSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<MeetingsSettings xmlns="http://soap.sforce.com/2006/04/metadata">
        <enableSalesforceMeetings>true</enableSalesforceMeetings>
        <enableSalesforceMeetingsSyncCheck>true</enableSalesforceMeetingsSyncCheck>
        <enableZoomVideoConference>false</enableZoomVideoConference>
</MeetingsSettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>MeetingsSettings</members>
        <name>Settings</name>
    </types>
    <version>51.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

MobileSettings

Represents an organization's mobile settings. This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

Note: MobileSettings on page 2025 is no longer available in API versions 25.0 and 26.0.

Declarative Metadata File Suffix and Directory Location

MobileSettings on page 2025 values are stored in a single file named `Mobile.settings` in the `settings` directory. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

Mobile settings are available in API version 27.0 and later.

Fields

Field	Field Type	Description
`chatterMobile` (Removed)	ChatterMobileSettings	The settings for devices running Chatter mobile. Removed in API version 46.0.
`dashboardMobile` (Deprecated)	DashboardMobileSettings	The settings for devices running the mobile dashboards app.
`enableImportContactFromDevice`	boolean	Indicates whether users can import contacts from their mobile device (`true`) or not (`false`). Available in API version 47.0 and later.
`enableLightningOnMobile` (Removed)	boolean	Indicates whether the org is enabled for the Salesforce mobile app. Available in API version 47.0 only. Removed in API version 48.0.
`enableNewSalesforceMobileAppForTablet` (Removed)	boolean	Indicates whether the org is enabled for the Salesforce mobile app tablet experience. Removed in API version 56.0.
`enableNewSalesforceMobileAppForTabletWideScreen` (Removed)	boolean	Indicates whether the org is enabled for the Salesforce mobile app widescreen tablet experience. Available in API version 52.0 through 55.0. Removed in API version 56.0.

Field	Field Type	Description
`enableOfflineDraftsEnabled`	boolean	Indicates whether users can create, edit, and delete records while offline in the Salesforce mobile app (`true`) or not (`false`). The default value is `true`. This option isn't available if `enableS1OfflinePref` is set to `false`. Available in API version 47.0 and later.
`enablePopulateNameManuallyInToday`	boolean	Indicates whether the user's name is shown on the Today page in the Salesforce mobile app (`true`) or not (`false`). The default value is `false`. Available in API version 47.0 and later.
`enableS1EncryptedStoragePref2`	boolean	Indicates whether the Salesforce mobile web uses secure and persistent browser caching to improve performance (`true`) or not (`false`). The default value is `true`. Available in API version 47.0 and later.
`enableS1OfflinePref`	boolean	Indicates whether users can access records offline in the Salesforce mobile app (`true`) or not (`false`). This option is set to `true` the first time someone in your org installs one of the Salesforce downloadable apps. Available in API version 47.0 and later. However, offline access isn't supported in all versions of the downloadable mobile apps. Users must have version 10.0 or later of the Salesforce for Android app or the Salesforce for iOS app. Offline access isn't available for the Salesforce mobile web.
`touchMobile` (Removed)	TouchMobileSettings	The settings for devices running Salesforce Touch. Removed in API version 46.0.

ChatterMobileSettings

These fields are removed in API version 46.0. Represents your organization's Chatter Mobile settings.

Field	Field Type	Description
`IPadAuthorized`	boolean	Indicates whether iPad devices are enabled for Chatter Mobile (`true`) or not (`false`).

Field	Field Type	Description
`IPhoneAuthorized`	boolean	Indicates whether iPhone devices are enabled for Chatter Mobile (`true`) or not (`false`).
`androidAuthorized`	boolean	Indicates whether Android devices are enabled for Chatter Mobile (`true`) or not (`false`).
`blackBerryAuthorized`	boolean	Indicates whether Blackberry devices are enabled for Chatter Mobile (`true`) or not (`false`).
`enableChatterMobile`	boolean	Indicates whether Chatter Mobile has been enabled for your organization (`true`) or not (`false`). Setting this field to `true` enables you to set all the other ChatterMobile settings. If you change this setting from `true` to `false`, and also try to change any of the other ChatterMobile settings, your deployment fails with an error.
`enablePushNotifications`	boolean	Indicates whether Chatter push notifications have been enabled for your organization (`true`) or not (`false`)
`sessionTimeout`	MobileSessionTimeout (enumeration of type string)	The length of time after which users without activity are prompted to log out or continue working. Valid values are: • `Never` • `OneMinute` • `FiveMinutes` • `TenMinutes` • `ThirtyMinutes`

DashboardMobileSettings

These fields are deprecated. Represents your organization's Mobile Dashboards iPad app settings.

Field	Field Type	Description
`enableDashboardIPadApp`	boolean	Indicates whether Mobile Dashboards iPad app has been enabled for your organization (`true`) or not (`false`).

TouchMobileSettings

These fields are removed in API version 46.0. Salesforce Touch has been upgraded to the Salesforce mobile app.

Field	Field Type	Description
`enableTouchBrowserIPad`	boolean	Indicates whether your organization has the Salesforce Touch mobile browser app enabled (`true`) or not (`false`).
`enableTouchAppIPad`	boolean	Indicates whether your organization has the Salesforce Touch downloadable app enabled (`true`) or not (`false`)

Declarative Metadata Sample Definition

Here's a sample `mobile.settings` metadata file.

```
<?xml version="1.0" encoding="UTF-8"?>
<MobileSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <enableImportContactFromDevice>true</enableImportContactFromDevice>
    <enableOfflineDraftsEnabled>true</enableOfflineDraftsEnabled>
    <enableS1EncryptedStoragePref2>true</enableS1EncryptedStoragePref2>
    <enableS1OfflinePref>true</enableS1OfflinePref>
</MobileSettings>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

MyDomainSettings

Represents your org's My Domain settings. With My Domain, you can include your company name in your URLs, for example, `https://yourcompanyname.my.salesforce.com`. This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

MyDomainSettings values are stored in a single file named `MyDomain.settings` in the `settings` directory. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

MyDomainSettings components are available in API version 47.0 and later.

Fields

Field Name	Field Type	Description
areLegacyRedirectsMaintained	boolean	Salesforce checks this field when this org is upgraded to Summer '25. If areLegacyRedirectsMaintained is false, Salesforce disables enableLegacyRedirections during the upgrade. If true, no change is made to enableLegacyRedirections during the upgrade. The default is false. This field has no impact after this org is on Summer '25. Available in API version 63.0 and later.
areLgcyRdirMaintainedWntr26	boolean	Salesforce checks this field when this org is upgraded to Winter '26. If areLgcyRdirMaintainedWntr26 is false, Salesforce disables enableLegacyRedirections during the upgrade. If true, no change is made to enableLegacyRedirections during the upgrade. The default is false. This field has no impact after this org is on Winter '26. Available in API version 64.0 and later.
canOnlyLoginWithMyDomainUrl	boolean	If true, users must use the org's My Domain login URL to log in. If false (default), users can also log in using the org's instance Salesforce URL, https://*InstanceName*.salesforce.com, and through the login URL https://login.salesforce.com. Admins can log in to a sandbox via the **Log In** action on the Sandboxes Setup page only when canOnlyLoginWithMyDomainUrl is false in the sandbox.
doesApiLoginRequireOrgDomain	boolean	If true, users must use the org's My Domain login URL to access the Salesforce API. If false (default), users can also access the Salesforce API using the generic Salesforce page, https://*InstanceName*.salesforce.com and through the login URL https://login.salesforce.com.
doesWarnOnForceComRedirect	boolean	Indicates whether users who visit a previous *.force.com site URL for this org see a message with the current URL during redirections (true) or not (false). The default is false. This field is applicable only when • Your org has a previous *.force.com URL associated with an Experience Cloud site or Salesforce Site. • redirectForceComSitesUrls is true This field is available in API version 59.0 and later.

Field Name	Field Type	Description
`doesWarnOnRedirect`	boolean	Indicates whether users who visit a previous My Domain URL see a message with the current URL during redirections (`true`) or not (`false`). The default is `false`. This field is applicable only when • Your org has a previous My Domain. For example, after an admin deploys a change to the My Domain. • The previous My Domain hasn't been removed via the Routing options on the My Domain Setup page. If the previous My Domain is removed, calls to URLs associated with that My Domain aren't redirected. • `redirectPriorMyDomain` is `true`. This field is available in API version 59.0 and later.
`domainPartition`	OrgDomainShard (enumeration of type string)	The partition for this org. When `none`, partitioned domains aren't enabled. Otherwise, My Domain hostnames include the partition value. For example, the format of a My Domain login hostname for a Developer Edition org with partitioned domains is *`MyDomainName`*`.develop.my.salesforce.com`. This field is read-only in the API. Possible values are: • `demo`—Used in demo orgs. Available in API version 60.0 and later. • `develop`—Used in Developer Edition orgs. Also used in patch orgs where partitioned domains were deployed before Winter '24. • `free`—Reserved for internal use. • `none`—Indicates that this org doesn't use partitioned domains. • `patch`—Used in patch orgs. Available in API version 59.0 and later. • `sandbox`—Used in sandboxes with enhanced domains. These orgs are always partitioned. • `scratch`—Used in scratch orgs. • `sfdctest`—Reserved for internal use. • `sfrestore`—Reserved for internal use. Available in API version 62.0 and later. • `trailblaze`—Used in Trailblazer Playgrounds. Partitioned domains require enhanced domains. Production orgs always have a value of `none`, and only the sandbox partition is available in Government Cloud—Defense orgs. Qualifying new orgs are partitioned by default and get the corresponding `domainPartition` value. You can't disable this feature in those orgs. Available in API version 55.0 and later.

Field Name	Field Type	Description
enableCrossDomainPreviewCookies	boolean	If isFirstPartyCookieUseRequired is true, indicates whether cross-domain cookies are allowed on the Experience Builder preview domain (true) or not (false). If isFirstPartyCookieUseRequired is false, this field has no effect. Available in API version 62.0 and later.
enableEdgeDuringRollout	boolean	Indicates whether Salesforce Edge Network is enabled in this org during the scheduled rollout. If true, Salesforce notifies admins by email before Salesforce Edge Network is enabled. If false, the Edge enablement is deferred. The default value is true. If useEdge is true, this field has no effect. Available in API version 58.0 and later.
enableExtendedRedirections	boolean	If Salesforce configured temporary extended redirections for select legacy hostnames for this org, indicates whether calls to those hostnames are redirected (true) or not (false). The default is true. If extended redirections aren't configured for this org, this field has no effect. Available in API version 62.0 and later.
enableLegacyRedirections	boolean	If redirectPriorMyDomain is true, indicates whether legacy hostnames for this org are redirected (true) or not (false). If any of these conditions are true, this field has no effect. • No legacy (non-enhanced) hostnames exist for the org • redirectPriorMyDomain is false • Salesforce stopped legacy redirections for this org For more information about the end of legacy hostname redirections, see Prepare for the End of Redirections for Non-Enhanced Domains in Salesforce Help. Available in API version 62.0 and later.
enableNativeBrowserForAuthOnAndroid	boolean	If true, use the native browser for authentication of Android mobile apps. Default is false.
enableNativeBrowserForAuthOnIos	boolean	If true, use the native browser for authentication of iOS mobile apps. Default is false.
instancedUrlRedirectHandling	OrgDomainRedirectOption (enumeration of type string)	Indicates how user visits to this org's instanced URL are handled. Possible values are: • Redirect—Redirect users to the equivalent page on this org's My Domain hostnames. • WarnOnRedirect—Display a brief warning message to users as they are redirected to the equivalent page on this org's My Domain hostnames.

Field Name	Field Type	Description
		• `NoRedirect`—Require users to use your My Domain hostnames when they view this org's pages. Visits to this page's instanced URL aren't redirected. For example, when a user visits `https://InstanceName.lightning.force.com/lightning/page/home`, this setting determines whether they're redirected to `https://MyDomainName.lightning.force.com/lightning/page/home`. The value of this field has no effect on user's ability to log in via the org's instanced URL. To control that behavior, use `canOnlyLoginWithMyDomainUrl`. Available in API version 59.0 and later.
`isFirstPartyCookieUseRequired`	boolean	Indicates whether the `SameSite=None` attribute is removed from Salesforce cookies (`true`) or not (`false`). In Salesforce orgs created in Summer '24 and later, the default is `true`. In all other orgs, the default is `false`. Available in API version 61.0 and later.
`isInstancedApiTrafficBlocked`	boolean	This field is reserved for future use. The default is `false`. Available in API version 64.0 and later.
`logRedirections`	boolean	If `true`, Salesforce produces a log for the Hostname Redirects event type when daily event logs are generated. The default is `false`. The Hostname Redirects event is free for all customers with a 24-hour data retention period. When `logRedirections` is `true`, this event is available in the API but not in the Event Monitoring Analytics app. You can also download the latest Hostname Redirects event log file through a button on the My Domain page. Available in API version 56.0 and later.
`myDomainName`	string	The subdomain name used in My Domain URLs for this org, such as `MyDomainName.my.salesforce.com` and `MyDomainName.lightning.force.com`. This field is read-only in the API. You can change your org's My Domain name from the My Domain Setup page. Available in API version 51.0 and later.
`myDomainSuffix`	OrgDomainProdSuffix (enumeration of type string)	The domain suffix for this org's My Domain login URL. This field is read-only in the API. Possible values are: • `CloudforceLimited`—`cloudforce.com` • `DatabaseLimited`—`database.com`

Field Name	Field Type	Description
		• `MySalesforce`—`my.salesforce.com` with enhanced domains • `MySalesforceLimited`—`my.salesforce.com` without enhanced domains • `OrgLevelCertificate`—`my-salesforce.com` • `OrgLevelCertificateLimited`—legacy version of `my-salesforce.com` that's noncompliant with browser settings that block third-party cookies • `Restricted1`—Reserved for future use. • `Restricted2`—Reserved for future use. Available in API version 51.0 and later.
`onboardCustomerSpecificSuffix`	boolean	This field is reserved for future use. The default is `false`. Available in API version 63.0 and later.
`redirectForceComSitesUrls`	boolean	If `true`, calls to URLs ending in `.force.com` that serve your Experience Cloud sites and Salesforce Sites are redirected to the corresponding current My Domain site URL. If `false`, these calls aren't redirected and the user gets a file not found (404) error. The default is `true`. This field is only applicable when • Enhanced domains are enabled. • Your org has a previous *.force.com URL associated with an Experience Cloud site or Salesforce Site. Available in API version 55.0 and later.
`redirectPriorMyDomain`	boolean	If `true`, calls to URLs associated with your previous My Domain name are redirected to the corresponding URL associated with your current My Domain. If `false`, these calls aren't redirected. When you deploy a new My Domain, this setting resets to its default, `true`. This field is only applicable when • Your org has a previous My Domain. For example, after an admin deploys a change to the My Domain. • The previous My Domain hasn't been removed via the Routing options on the My Domain Setup page. If the previous My Domain is removed, calls to URLs associated with that My Domain aren't redirected. Available in API version 54.0 and later.
`use3rdPartyCookieBlockingCompatibleHostnames`	boolean	Indicates whether the org's instance name is included in Visualforce URLs when third-party cookies are blocked (`true`) or not (`false`). This field has a default value of `true`. Setting this field `true` prevents potential issues loading Visualforce pages with stabilized URLs.

Field Name	Field Type	Description
		Only applicable when `useStabilizedMyDomainHostnames` is set to `true` and `myDomainSuffix` is set to `MySalesforceLimited`, `CloudforceLimited`, or `DatabaseLimited`. Available in API version 51.0 and later.
`useEdge`	boolean	Indicates whether this org's qualifying My Domain URLs are routed through Salesforce Edge Network (`true`) or not (`false`). This field has a default value of `true`. This field is read-only in the API. If your org can use Salesforce Edge Network, you can enable this setting from the My Domain Setup page. After this field is set to `true` from Setup, it can't be set to `false`. Available in API version 51.0 and later.
`useEnhancedDomainsInSandbox`	boolean	This field corresponds to the previous My Domain setting, Use enhanced domains by default in new and refreshed sandboxes, that was removed in Summer '23. Previously, in API versions 55.0 to 57.0, if enhanced domains weren't enabled, this field indicated whether new and refreshed sandboxes created from this org used enhanced domains by default (`true`) or not (`false`), and the default value was `true`. As of API version 58.0, this field's value is always `true`, regardless of the value that you set. Changing its value has no effect on Salesforce, even if it reads `false`.
`useStabilizedMyDomainHostnames`	boolean	Indicates whether the instance name is hidden in My Domain URLs for Visualforce, Experience Builder, Site.com Studio, and content files (`true`) or not (`false`). This field has a default value of `true`. For example, *`MyDomainName--PackageName`*`.na44.visual.force.com` becomes *`MyDomainName--PackageName`*`.visualforce.com` when this field is set to `true`. Only applicable when `myDomainSuffix` is set to `MySalesforceLimited`, `CloudforceLimited`, or `DatabaseLimited`.
`useStabilizedSandboxMyDomainHostnames`	boolean	This field corresponds to the Stabilize the Hostname for My Domain URLs in Sandboxes release update, which was enforced in Summer '20. When `true`, the instance name is hidden in My Domain URLs for sandboxes orgs. For example, *`MyDomainName`*`--test.cs5.my.salesforce.com` became *`MyDomainName`*`--test.my.salesforce.com`. As of API version 49.0, this field's value is always `true`, regardless of the value that you set. Changing its value has no effect on Salesforce, even if it reads `false`.

Field Name	Field Type	Description
		This change applies retroactively back to API version 47.0, when this field was first introduced. Previously, in API version 47.0 to 49.0, this field indicated whether the instance name was hidden in My Domain URLs for sandboxes orgs (`true`) or not (`false`), and the field's default value was `false`. Now, in all API versions, this field's value is always `true`, even if it reads `false`.

Declarative Metadata Sample Definition

The following is an example of a MyDomainSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<MyDomainSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <areLegacyRedirectsMaintained>false</areLegacyRedirectsMaintained>
    <areLgcyRdirMaintainedWntr26>false</areLgcyRdirMaintainedWntr26>
    <canOnlyLoginWithMyDomainUrl>false</canOnlyLoginWithMyDomainUrl>
    <doesApiLoginRequireOrgDomain>false</doesApiLoginRequireOrgDomain>
    <doesWarnOnForceComRedirect>false</doesWarnOnForceComRedirect>
    <doesWarnOnRedirect>true</doesWarnOnRedirect>
    <domainPartition>none</domainPartition>
    <enableCrossDomainPreviewCookies>true</enableCrossDomainPreviewCookies>
    <enableEdgeDuringRollout>true</enableEdgeDuringRollout>
    <enableExtendedRedirections>false</enableExtendedRedirections>
    <enableLegacyRedirections>false</enableLegacyRedirections>
    <enableNativeBrowserForAuthOnAndroid>false</enableNativeBrowserForAuthOnAndroid>
    <enableNativeBrowserForAuthOnIos>false</enableNativeBrowserForAuthOnIos>
    <instancedUrlRedirectHandling>false</instancedUrlRedirectHandling>
    <isFirstPartyCookieUseRequired>true</isFirstPartyCookieUseRequired>
    <isInstancedApiTrafficBlocked>true</isInstancedApiTrafficBlocked>
    <logRedirections>true</logRedirections>
    <myDomainName>mycompany</myDomainName>
    <myDomainSuffix>MySalesforce</myDomainSuffix>
    <onboardCustomerSpecificSuffix>false</onboardCustomerSpecificSuffix>
    <redirectForceComSitesUrls>true</redirectForceComSitesUrls>
    <redirectPriorMyDomain>true</redirectPriorMyDomain>

<use3rdPartyCookieBlockingCompatibleHostnames>true</use3rdPartyCookieBlockingCompatibleHostnames>

    <useEdge>true</useEdge>
    <useEnhancedDomainsInSandbox>true</useEnhancedDomainsInSandbox>
    <useStabilizedMyDomainHostnames>true</useStabilizedMyDomainHostnames>
    <useStabilizedSandboxMyDomainHostnames>true</useStabilizedSandboxMyDomainHostnames>
</MyDomainSettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>MyDomain</members>
        <name>Settings</name>
```

```
    </types>
    <version>64.0</version>
</Package>
```

MfgServiceConsoleSettings

Represents the settings to access the Service Console for Manufacturing.This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for more details.

File Suffix and Directory Location

MfgServiceConsoleSettings values are stored in the `MfgServiceConsole.settings` file in the `settings` directory.

Version

MfgServiceConsoleSettings components are available in API version 56.0 and later.

Special Access Rules

To use this metadata type, your Salesforce org must have the Manufacturing Cloud license.

Fields

Field Name	Field Type	Description
`enableMfgServiceConsole`	boolean	Indicates whether Service Console for Manufacturing is enabled in your org (`true`) or not `false`). Note: By default, Service Console for Manufacturing is disabled.

Declarative Metadata Sample Definition

The following is an example of a MfgServiceConsoleSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<MfgServiceConsoleSettings
 xmlns="http://soap.sforce.com/2006/04/metadata">
 <enableMfgServiceConsole>true</enableMfgServiceConsole>
</MfgServiceConsoleSettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package
 xmlns="http://soap.sforce.com/2006/04/metadata">
 <types>
```

```
  <members>MfgServiceConsole</members>
  <name>Settings</name>
 </types>
 <version>56.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

NameSettings

Enables or disables the formal name, middle name, and suffix attributes for these person objects: Contact, Lead, Person Account, and User. This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

NameSettings values are stored in a single file named `Name.settings` in the `settings` folder. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

NameSettings components are available in API version 31.0 and later.

Fields

Field Name	Field Type	Description
`enableInformalName`	boolean	Reserved for internal use. Available in API version 48.0 and later.
`enableMiddleName`	boolean	Indicates whether middle names are enabled (`true`) or disabled (`false`) for person objects.
`enableNameSuffix`	boolean	Indicates whether suffixes are enabled (`true`) or disabled (`false`) for person objects.

Declarative Metadata Sample Definition

This example shows a NameSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<NameSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <enableMiddleName>true</enableMiddleName>
    <enableNameSuffix>false</enableNameSuffix>
</NameSettings>
```

This example `package.xml` manifest references the NameSettings definitions.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>Name</members>
        <name>Settings</name>
    </types>
    <version>31.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

NotificationsSettings

Represents an organization's mobile settings.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

Declarative Metadata File Suffix and Directory Location

NotificationsSettings values are stored in a single file named `Notifications.settings` in the `settings` directory. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

Mobile settings are available in API version 46.0 and later.

Fields

Field	Field Type	Description
`enableActivityReminderBrowserNotifs`	boolean	Reserved for internal use.
`enableMobileAppPushNotifications`	boolean	Indicates whether mobile push notifications are enabled.
`enableNotifications`	boolean	Indicates whether notifications are enabled.

Declarative Metadata Sample Definition

This is a sample `notifications.settings` metadata file.

```
<NotificationsSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <enableMobileAppPushNotifications>true</enableMobileAppPushNotifications>
```

```
    <enableNotifications>true</enableNotifications>
</NotificationsSettings>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

OauthOidcSettings

Represents org settings for disabling OAuth OpenID Connect authorization flows.

Parent Type and Manifest Access

This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all the settings metadata types for the org are accessed using the "Settings" name. See Settings for more details.

File Suffix and Directory Location

`OauthOidcSettings` values are stored in the `OauthOidc.settings` file in the `settings` folder. The `.settings` files are different from other named components, because there is only one settings file for each settings component.

Version

OauthOidcSettings is available in API version 56.0 and later.

Special Access Rules

There are no additional access requirements that are specific to this type.

Fields

Field Name	Description
`blockOAuthUnPwFlow`	**Field Type** boolean **Description** Indicates whether the username-password flow is blocked (`true`) or not blocked (`false`). The default value is `false`.
`blockOAuthUsrAgtFlow`	**Field Type** boolean **Description** Indicates whether the user-agent flow is blocked (`true`) or not blocked (`false`). The default value is `false`.

Field Name	Description
`enableHdlessFgtPswFlow`	**Field Type** boolean **Description** For internal use only.
`isPkceRequired`	**Field Type** boolean **Description** Indicates whether the OAuth 2.0 Proof Key for Code Exchange (PKCE) security extension is required for variations of the OAuth authorization code flow that access this org (`true`) or not (`false`). This setting requires PKCE for all supported variations of the authorization code flow, including the web server flow, the Authorization Code and Credentials Flow, and their derivatives. The default value is `false`. This field is available in API version 59.0 and later.
`oAuthCdCrdtFlowEnable`	**Field Type** boolean **Description** Indicates whether the Authorization Code and Credentials Flow is enabled (`true`) or blocked (`false`). The default value is `false`.

Declarative Metadata Sample Definition

The following is an example of the OauthOidcSettings file.

```
<?xml version="1.0" encoding="UTF-8"?>
<OauthOidcSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <blockOAuthUnPwFlow>true</blockOAuthUnPwFlow>
    <blockOAuthUsrAgtFlow>true</blockOAuthUsrAgtFlow>
    <oAuthCdCrdtFlowEnable>false</oAuthCdCrdtFlowEnable>
</OauthOidcSettings>
```

Example Package Manifest

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
  <types>
    <members>OauthOidc</members>
    <name>Settings</name>
  </types>
  <version>56.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ObjectHierarchyRelationship

Represents an organization's custom field mappings for sales agreement conversion. Fields can be mapped from Opportunity and Quotes to SalesAgreement and SalesAgreementProduct.

This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

ObjectHierarchyRelationship components have the suffix `ObjectHierarchyRelationship.settings` and are stored in the `ObjectHierarchyRelationship` folder.

Version

ObjectHierarchyRelationship components are available in API version 51.0 and later.

Fields

Field Name	Description
`childObjectMapping`	**Field Type** ObjectMapping **Description** Set of `inputObject`, `mappingFields`, and `outputObject` entries. For example, fields from the input object of OpportunityLineItem maps to fields of the output object of SalesAgreementProduct.
`childObjectMappingId`	**Field Type** String **Description** The ID of the child object mapping record. This field is available in API version 56.0 and later.
`inputObjRecordsGrpFieldName`	**Field Type** string **Description** The field name in the input object used to group the records. This field is available in API version 55.0 and later.

Field Name	Description
`mappingType`	**Field Type** ObjHierarchyMappingType (enumeration of type string) **Description** Specifies the type of relationship between two objects. This field is available in API version 55.0 and later. Valid values are: • `ChildToChild` • `ParentToChild` • `ParentToParent` • `Support`
`masterLabel`	**Field Type** string **Description** Label name of the mapping definition.
`outputPntRelationshipFieldName`	**Field Type** string **Description** The field name that defines the relationship between a parent and child for the output object. This field is available in API version 55.0 and later.
`parentObjectMapping`	**Field Type** ObjectMapping **Description** Required. Set of `inputObject`, `mappingFields`, and `outputObject` entries. For example, fields from the input object of Opportunity maps to fields of the output object of SalesAgreement.
`parentObjectMappingId`	**Field Type** string **Description** The ID of the parent object mapping record. This field is available in API version 56.0 and later.
`parentRecord`	**Field Type** string **Description** The parent record for this object hierarchy relationship. This field is available in API version 55.0 and later.

Field Name	Description
`parentRelationshipFieldName`	**Field Type** string **Description** Name of the field that defines the relationship between the parent and child.
`sourceReferenceRelaFieldName`	**Field Type** string **Description** The field name in an object that's used to define the relationship between a source and reference object. This field is available in API version 56.0 and later.
`usageType`	**Field Type** MappingUsageType (enumeration of type string) **Description** Required. Name of the usage type of an object hierarchy relationship. Valid value is: • `ConvertToSalesAgreement` • `CLMFieldMapping` • `EligibleProgramRebateType` • `MapJournalToMemberAggregate` • `TransformationMapping`

ObjectMapping

Represents a set of `inputObject`, `mappingFields`, and `outputObject` entries.

Fields

Field Name	Description
`inputObject`	**Field Type** string **Description** Required. Name of the input object type containing the source fields for mapping. For example, Opportunity or OpportunityLineItem.
`mappingFields`	**Field Type** ObjectMappingField

Field Name	Description
	Description Mapping of source object input fields to target object for SalesAgreement and SalesAgreementProduct.
`outputObject`	**Field Type** string **Description** Required. Name of the output object type receiving data conversion. For example, SalesAgreement or SalesAgreementProduct.

ObjectMappingField

Represents a set of `inputField` and `outputField` entries.

Fields

Field Name	Description
`inputField`	**Field Type** string **Description** Required. Field in the object specified by the `inputObject` field in ObjectMapping on page 2043. This field is mapped to the field in `outputField`, which is a field in the object specified by the `outputObject` field in ObjectMapping on page 2043.
`outputField`	**Field Type** string **Description** Required. Field in the object specified by the `outputObject` field in ObjectMapping on page 2043. This field is mapped to the field name in `inputField`, which is a field in the object specified by the `inputObject` field in ObjectMapping on page 2043.

Declarative Metadata Sample Definition

The following is an example of a ObjectHierarchyRelationship component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ObjectHierarchyRelationship xmlns="http://soap.sforce.com/2006/04/metadata">
```

```
  <parentObjectMapping>
    <inputObject>Opportunity</inputObject>
    <mappingFields>
      <inputField>Name</inputField>
      <outputField>Name</outputField>
    </mappingFields>
    <mappingFields>
      <inputField>CloseDate</inputField>
      <outputField>StartDate</outputField>
    </mappingFields>
    <mappingFields>
      <inputField>Account</inputField>
      <outputField>Account</outputField>
    </mappingFields>
    <mappingFields>
      <inputField>Pricebook2</inputField>
      <outputField>Pricebook</outputField>
    </mappingFields>
    <outputObject>SalesAgreement</outputObject>
  </parentObjectMapping>

  <childObjectMapping>
    <inputObject>OpportunityLineItem</inputObject>
    <mappingFields>
      <inputField>Id</inputField>
      <outputField>Name</outputField>
    </mappingFields>
    <mappingFields>
      <inputField>UnitPrice</inputField>
      <outputField>SalesPrice</outputField>
    </mappingFields>
    <mappingFields>
      <inputField>PricebookEntry</inputField>
      <outputField>PricebookEntry</outputField>
    </mappingFields>
    <mappingFields>
      <inputField>Quantity</inputField>
      <outputField>InitialPlannedQuantity</outputField>
    </mappingFields>
    <outputObject>SalesAgreementProduct</outputObject>
  </childObjectMapping>

  <masterLabel>ObjectHierarchyRelationship</masterLabel>
  <usageType>ConvertToSalesAgreement</usageType>
  <parentRelationshipFieldName>Opportunity</parentRelationshipFieldName>
  <outputPntRelationshipFieldName>SalesAgreement</outputPntRelationshipFieldName>
  <parentRecord></parentRecord>
  <inputObjRecordsGrpFieldName>Account</inputObjRecordsGrpFieldName>
  <mappingType>ParentToParent</mappingType>

</ObjectHierarchyRelationship>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
      <members>*</members>
      <name>ObjectHierarchyRelationship</name>
    </types>
    <version>51.0</version>
</Package>
```

Usage

Use the `deploy()` call to deploy metadata with a .zip file. Every .zip file contains a project manifest, a file that's named package.xml, and a set of directories that contain the components. The manifest file defines the components that you're trying to retrieve or deploy in the .zip file. The manifest also defines the API version that's used for the deployment or retrieval. For more information on the .zip file, deploying, and retrieving metadata, see Deploying and Retrieving Metadata with the Zip File. You can also deploy and retrieve the metadata API using Postman.

Ensure you map all the required fields for sales agreement conversion.

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ObjectLinkingSettings (Beta)

Represents the channel-object linking settings for an org. This type extends the Metadata metadata type and inherits its `fullName` field.

Note: As a beta feature, Channel-Object Linking is a preview and isn't part of the "Services" under your Main Services Agreement with Salesforce. Use this feature at your sole discretion, and make your purchase decisions only on the basis of generally available products and features. Salesforce doesn't guarantee general availability of this feature within any particular time frame or at all, and we can discontinue it at any time. This feature is for evaluation purposes only, not for production use. It's offered as is and isn't supported, and Salesforce has no liability for any harm or damage arising out of or in connection with it. All restrictions, Salesforce reservation of rights, obligations concerning the Services, and terms for related Non-Salesforce Applications and Content apply equally to your use of this feature. For information on enabling this feature, contact Salesforce.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

ObjectLinkingSettings values are stored in the `ObjectLinking.settings` file in the `settings` folder. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

ObjectLinkingSettings components are available in API version 47.0 and later.

Fields

Field Name	Field Type	Description
`enableObjectLinking`	boolean	Indicates whether Channel-Object Linking is enabled, allowing you to link channel interactions to objects such as Contacts. The default value is `false`.

Declarative Metadata Sample Definition

The following is an example of an ObjectLinkingSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ObjectLinkingSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <enableObjectLinking>true</enableObjectLinking>
</ObjectLinkingSettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>ObjectLinking</members>
        <name>Settings</name>
    </types>
    <version>47.0</version>
</Package>
```

OmniChannelSettings

Represents the Omni-Channel settings for an org.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

OmniChannelSettings values are stored in the `OmniChannel.settings` file in the `settings` folder.

Version

OmniChannelSettings components are available in API version 44.0 and later.

Fields

Field Name	Field Type	Description
`enableOmniAutoLoginPrompt`	boolean	Indicates whether to display a login confirmation upon loading a console with Omni-Channel. The default value is `false`.

Field Name	Field Type	Description
		When `true`, the console displays a prompt before logging into Omni-Channel when an agent opens another Omni-Channel console in a different tab or window or refreshes the current tab. The agent is logged out of Omni-Channel on other consoles and any ongoing conversations are ended. Available in API version 47.0 and later.
`enableOmniChannel`	boolean	Indicates whether Omni-Channel is enabled, giving you access to the objects required to set up the feature in your org. The default value is `false`.
`enableOmniSecondaryRoutingPriority`	boolean	Indicates whether Secondary Routing Priority is enabled in your org. The default value is `false`. Available in API version 47.0 and later.
`enableOmniSkillsRouting`	boolean	Indicates whether skills-based routing is enabled in your org. The default value is `false`.

Declarative Metadata Sample Definition

The following is an example of a OmniChannelSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<OmniChannelSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <enableOmniChannel>true</enableOmniChannel>
    <enableOmniAutoLoginPrompt>true</enableOmniAutoLoginPrompt>
    <enableOmniSecondaryRoutingPriority>true</enableOmniSecondaryRoutingPriority>
    <enableOmniSkillsRouting>true</enableOmniSkillsRouting>
</OmniChannelSettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>OmniChannel</members>
        <name>Settings</name>
    </types>
    <version>44.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

OpportunityInsightsSettings

Represents an org's Einstein Opportunity Insights settings. This setting controls features that give you relevant updates about your opportunities.

Note: This metadata type has been deprecated as of API version 59.0.

This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

OpportunityInsightsSettings values are stored in the `OpportunityInsights.settings` file in the `settings` folder. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

OpportunityInsightsSettings is available in API versions 48.0 to 58.0.

Fields

Field Name	Field Type	Description
`enableOpportunityInsights`	boolean	Indicates whether Einstein Opportunity Insights is enabled (`true`) or not (`false`). The default value is `false`.

Declarative Metadata Sample Definition

The following is an example of the OpportunityInsights.settings file:

```
<?xml version="1.0" encoding="UTF-8"?>
<OpportunityInsightsSettings xmlns="http://soap.sforce.com/2006/04/metadata">
 <enableOpportunityInsights>true</enableOpportunityInsights>
</OpportunityInsightsSettings>
```

Example Package Manifest

The following is an example package manifest used to deploy or retrieve the OpportunityInsights settings metadata:

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
 <types>
  <members>OpportunityInsights</members>
  <name>Settings</name>
 </types>
 <version>29.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

OpportunitySettings

Represents org preferences for features such as automatic opportunity updates and similar-opportunity filters.

This type extends the Metadata metadata type and inherits its `fullName` field.

Use opportunity settings to control the actions that users can perform on their opportunities.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

Opportunities values are stored in a single file named `Opportunity.settings` in the `settings` directory of the corresponding package directory.

Version

OpportunitySettings is available in API version 28.0 and later.

Fields

Field Name	Field Type	Description
`autoActivateNewReminders`	boolean	Automatically uses scheduled updates for new opportunities.
`customizableProductSchedulesEnabled`	boolean	Lets Salesforce admins customize product schedules by using custom fields, validation rules, and Apex triggers on the LineItemSchedule object. This field is available in API version 46.0 and later. If customizable product schedules are enabled, you can use custom fields in default schedules and customize their layout, but Apex triggers or validation rules that you apply to default schedules are bypassed.
`doesEnforceStandardOpportunitySaveLogic`	boolean	Enforces standard validation and triggers for opportunity products and opportunity product schedules. Default value is `true`. Can't be set to `false`. Available in API version 47.0 and later.
`enableExpandedPipelineInspectionSetup`		Displays a Pipeline Inspection setup page to Salesforce admins with all setup steps for enabling and configuring the feature. The set up also includes historical trending. The default value is false. Available in API version 52.0 and later.
`enableFindSimilarOpportunities`	boolean	Lets users see related or similar existing opportunities.
`enableForecastCategoryMetrics`	boolean	Lets users see single and cumulative forecast category rollups over a selected period. Applies to the following categories: Best Case, Closed Lost, Closed Won, Commit, Most Likely, Open Pipeline, and Total. Default value is `true`. Available in API version 57.0 and later.

Field Name	Field Type	Description
enableOpportunityFieldHistoryTracking	boolean	Enables history tracking for the opportunity field. For more information, see "Field History Tracking" in Salesforce Help. Default value is `true`. Available in API version 47.0 and later.
enableOpportunityInsightsInMobile	boolean	Deprecated in API version 59.0 and later because the feature is no longer available. Indicates whether a user can see Einstein Opportunity Insights on their mobile device `(true)` or not `(false)`. Einstein Opportunity Insights includes predictions about which deals are likely to be won, reminders to follow up, and notifications when key moments in a deal take place. Available in API version 47.0 to 58.0.
enableOpportunityTeam	boolean	Lets users associate team members with opportunities.
enablePipelineChangesMetrics	boolean	Lets users see net change to the pipeline (positive or negative) contributed by each deal within a selected timeframe. Applies to the following categories: Open Pipeline, New, Won, Increased, Moved In, Moved Out, Decreased, Lost, and Overdue. Default value is `true`. Available in API version 57.0 and later.
enablePipelineInspection	boolean	Enables the Pipeline Inspection feature in the Opportunity tab. Also enables historical trending for opportunities, if the org has the historical trending org perm. Pipeline Inspection is a consolidated view of pipeline metrics, corresponding opportunities, and highlights of recent opportunity changes and insights. The default value is false. Also enables historical trending for opportunities, if historical trending isn't already enabled. To use Pipeline Inspection, additional configuration in Setup is required. Available in API version 52.0 and later.
enablePipelineInspectionFlow	boolean	Enables the Pipeline Inspection Flow Chart in the Opportunity tab. This chart shows Pipeline Inspection users the changes to opportunities in different forecast categories over time. Users can filter results to see the data that's most useful to them. To use this feature, access to Revenue Insights is required. Available in API version 54.0 and later.
enablePipelineInspectionSingleCategoryRollup	boolean	Indicates that Pipeline Inspection metrics display as single forecast categories `(true)`, or multiple categories rolled up `(false)`. The default value is `(false)`. To use this feature, Pipeline Inspection configuration in Setup is required. Available in API version 55.0 and later.

Field Name	Field Type	Description
`enableRevenueInsights`	boolean	Sets up Revenue Insights dashboards and installs the related CRM Analytics app. The dashboards give users access to sales performance, pipeline, and forecasting reports and analytics. Revenue Insights is part of Revenue Intelligence, which is available for an additional cost. Available in API version 54.0 and later.
`enableServiceCaseInsights`	boolean	Indicates whether insights based on service cases are enabled `(true)` or not `(false)` in Pipeline Inspection. The default value is `(false)`. Available in API version 55.0 and later.
`enableUpdateReminders`	boolean	Lets users enable automatic, scheduled updates on opportunities.
`findSimilarOppFilter`	FindSimilarOppFilter on page 2052	Defines parameters for similar opportunities.
`oppAmountDealMotionEnabled`	boolean	Indicates whether deal change highlights are enabled for opportunity amounts `(true)` or not `(false)`. The default value is `(true)`. Available in API version 50.0 and later.
`oppCloseDateDealMotionEnabled`	boolean	Indicates whether deal change highlights are enabled for opportunity close dates `(true)` or not `(false)`. The default value is `(true)`. Available in API version 50.0 and later.
`promptToAddProducts`	boolean	Prompts users to add related products to an opportunity.
`pushCountEnabled`	boolean	Indicates whether the Push Count field is visible to users in opportunity list views and Pipeline Inspection views `(true)` or not (`false`). The default value is (`true`). Available in API version 56.0 and later.
`simpleOppCreateFromContact`	boolean	Indicates whether you can create an opportunity with prefilled information (such as the contact's account) from the Global Actions menu while viewing a contact `(true)` or not `(false)`. Available in API version 54.0 and later.
`simpleOppCreateFromEvent`	boolean	Indicates whether you can create an opportunity with prefilled information (such as the event's account) from the Global Actions menu while viewing an event `(true)` or not `(false)`. Available in API version 54.0 and later.

FindSimilarOppFilter

Defines whether to match by entire columns or fields.

Field	Field Type	Description
similarOpportunitiesDisplayColumns	string	The columns to compare.
similarOpportunitiesMatchFields	string	The fields to compare.

Declarative Metadata Sample Definition

The following is an example of the package file.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>Opportunity</members>
        <name>Settings</name>
    </types>
    <version>28.0</version>
</Package>
```

The package file references the following Opportunity.settings file.

```
<?xml version="1.0" encoding="UTF-8"?>
<OpportunitySettings xmlns="http://soap.sforce.com/2006/04/metadata">
  <autoActivateNewReminders>true</autoActivateNewReminders>
  <customizableProductSchedulesEnabled>false</customizableProductSchedulesEnabled>
  <doesEnforceStandardOpportunitySaveLogic>true</doesEnforceStandardOpportunitySaveLogic>

  <enableFindSimilarOpportunities>true</enableFindSimilarOpportunities>
  <enableForecastCategoryMetrics>true</enableForecastCategoryMetrics>
  <enablePipelineChangesMetrics>true</enablePipelineChangesMetrics>
  <findSimilarOppFilter>
    <similarOpportunitiesMatchFields>OPPORTUNITY.Account</similarOpportunitiesMatchFields>


<similarOpportunitiesMatchFields>OPPORTUNITY.OpportunityCompetitors</similarOpportunitiesMatchFields>

    <similarOpportunitiesMatchFields>CustomField__c</similarOpportunitiesMatchFields>
   <similarOpportunitiesDisplayColumns>CustomField__c</similarOpportunitiesDisplayColumns>

  </findSimilarOppFilter>
  <enableOpportunityFieldHistoryTracking>true</enableOpportunityFieldHistoryTracking>
  <enableOpportunityInsightsInMobile>false</enableOpportunityInsightsInMobile>
  <enableOpportunityTeam>true</enableOpportunityTeam>
  <enableUpdateReminders>true</enableUpdateReminders>
  <promptToAddProducts>true</promptToAddProducts>
..<oppAmountDealMotionEnabled>true</oppAmountDealMotionEnabled>
..<oppCloseDateDealMotionEnabled>true</oppCloseDateDealMotionEnabled>
</OpportunitySettings>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

OpportunityScoreSettings

Represents an org's Einstein Opportunity Scoring settings, such as whether or not Einstein Opportunity Scoring is enabled. Einstein Opportunity Scoring helps determine the likelihood of an opportunity being won. This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

OpportunityScoreSettings values are stored in the `OpportunityScore.settings` file in the `settings` folder. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

OpportunityScoreSettings is available in API versions 49.0 and later.

Fields

Field Name	Field Type	Description
`enableOpportunityScoring`	boolean	Indicates whether Einstein Opportunity Scoring is enabled (`true`) or not (`false`). The default value is `false`.

Declarative Metadata Sample Definition

The following is an example of the OpportunityScore.settings file:

```
<?xml version="1.0" encoding="UTF-8"?>
<OpportunityScoreSettings xmlns="http://soap.sforce.com/2006/04/metadata">
 <enableOpportunityScoring>true</enableOpportunityScoring>
</OpportunityScoreSettings>
```

Example Package Manifest

The following is an example package manifest used to deploy or retrieve the OpportunityScore settings metadata:

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
 <types>
  <members>OpportunityScore</members>
  <name>Settings</name>
 </types>
```

```
 <version>49.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

OrderManagementSettings

Represents options for the Salesforce Order Management product. This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

OrderManagementSettings values are stored in the `OrderManagement.settings` file in the `settings` directory. The `.settings` files are different from other named components because there is only one settings file for each settings component.

Version

Order Management settings are available in API version 48 and later.

Special Access Rules

This metadata type is only accessible by developers and customers using Salesforce Order Management.

Fields

Field Name	Field Type	Description
`deliveryEstimationEnabled`	boolean	Specifies whether delivery estimation is active (`true`) or not (`false`). The default value is `false`. Available in API version 62.0 and later.
`enableB2CIntegration`	boolean	Indicates whether Order Management is allowed to accept order data from B2C Commerce (`true`) or not (`false`). The default value is `false`.
`enableDuplicateManagement`	boolean	Indicates whether the Order Management B2C Commerce Integration applies the Salesforce org's duplicate and matching rules for Accounts, Contacts, and Person Accounts to shopper records (`true`) or not (`false`). The default value is `false`. The Setup toggle label is **B2C Integration Data Matching Rules**. Available in API version 53.0 and later.
`enableHighScaleOrders`	boolean	Indicates whether the Order Management B2C Commerce Integration uses the High Scale Orders feature (`true`) or the original order ingestion

Field Name	Field Type	Description
		system (`false`). The default value is `false`. Available in API version 56.0 and later.
`enableOrderManagement`	boolean	Indicates whether Order Management features are enabled (`true`) or not (`false`). The default value is `false`.
`enablePersonAccountsForShoppers`	boolean	Indicates whether Order Management represents each shopper with a Person Account (`true`) or a normal Account and a Contact (`false`). The default value is `false`. Available in API version 49.0 and later.

Declarative Metadata Sample Definition

The following is an example of an OrderManagementSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<OrderManagementSettings xmlns="http://soap.sforce.com/2006/04/metadata"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <enableOrderManagement>true</enableOrderManagement>
    <deliveryEstimationEnabled>false</deliveryEstimationEnabled>
    <enableB2CIntegration>true</enableB2CIntegration>
    <enableDuplicateManagement>true</enableB2CIntegration>
    <enableHighScaleOrders>false</enableB2CSelfService>
    <enablePersonAccountsForShoppers>true</enablePersonAccountsForShoppers>
    </OrderManagementSettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>OrderManagement</members>
        <name>Settings</name>
    </types>
    <version>49.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

OrderSettings

Represents order settings.

This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

There is one OrderSettings component in a file named `Order.settings` in the `settings` folder. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

OrderSettings components are available in API version 30.0 and later.

Fields

Field Name	Field Type	Description
`enableEnhancedCommerceOrders`	boolean	Indicates whether enhanced commerce orders are enabled for the org (`true`) or not (`false`). This preference is available only in orgs with the Salesforce Order Management license. Default value is `false`. Available in API versions 48.0 and later.
`enableNegativeQuantity`	boolean	Indicates whether users in the org can add order products with quantities of less than zero (`true`) or not (`false`). To enable this preference, `enableOrders` must be set to `true`.
`enableOptionalPricebook`	boolean	Indicates whether users in the org can create orders without price books (`true`) or not (`false`). For more information, see Enable Orders Without Price Books in Salesforce Help.
`enableOrderEvents`	boolean	Indicates whether order events are enabled for the org (`true`) or not (`false`). For more information, see OrderStatusChangedEvent in the Platform Events Developer Guide.
`enableOrders`	boolean	Indicates whether orders are enabled for the org (`true`) or not (`false`).
`enableOrderWithMultiplePriceBooks`	boolean	Indicates whether users in the org can create orders containing order items that refer to multiple price books (`true`) or not (`false`). Available in API version 60.0 and later. To enable this preference, `enableOrders` and `enableEnhancedCommerceOrders` must be set to `true`. This field helps to offer different pricing structures for various customer segments, regions, or promotional periods.
`enableReductionOrders`	boolean	Indicates whether reduction orders are enabled for the org (`true`) or not (`false`). For more information, see Reduction Orders in Salesforce Help. To enable this preference, `enableOrders` must be set to `true`.
`enableZeroQuantity`	boolean	Indicates whether users in the org can add order products with quantities of zero (`true`) or not (`false`). Default value is `false`. To enable this preference, `enableOrders` must be set to `true`. Available in API version 42.0 and later.

Declarative Metadata Sample Definition

This is a sample OrderSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<OrderSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <enableNegativeQuantity>false</enableNegativeQuantity>
    <enableZeroQuantity>false</enableZeroQuantity>
    <enableOrders>true</enableOrders>
    <enableReductionOrders>true</enableReductionOrders>
    <enableEnhancedCommerceOrders>true</enableEnhancedCommerceOrders>
    <enableOptionalPricebook>false</enableOptionalPricebook>
    <enableOrderEvents>false</enableOrderEvents>
    <enableOrderWithMultiplePriceBooks>false</enableOrderWithMultiplePriceBooks>
</OrderSettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>Order</members>
        <name>Settings</name>
    </types>
    <version>64.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

OrgPreferenceSettings

Removed in API version 48.0. Represents the unique org preference settings in a Salesforce org.

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

OrgPreferenceSettings values are stored in the `OrgPreference.settings` file in the settings directory. The `.settings` files are different from other named components because there is only one settings file for each settings component.

Version

OrgPreferenceSettings components are available in API versions 37.0 to 47.0.

OrgPreferenceSettings is deprecated in API version 47.0 and removed in API version 48.0. In API version 47.0, most of the settings supported in the `preferences` field were made available in the form of Boolean fields on other Settings types. For example, in API version 47.0 and later, you can enable and disable the `CompileOnDeploy` preference by using the `enableCompileOnDeploy` field on the ApexSettings type.

Fields

Field Name	Field Type	Description
`preferences`	OrganizationSettingsDetail[]	The preferences associated with the org settings. In the following list of preferences, click hyperlinked preference names to go to the topic for the Settings type that contains that preference. If there is no link, the preference hasn't been moved to another Settings type. • `AnalyticsSharingEnable` (available in API version 40.0 and later) • `ApexApprovalLockUnlock` • `AsyncSaveEnabled` (available in API versions 40.0 to 46.0) • `ChatterEnabled` • `CompileOnDeploy` (available in API version 43.0 and later) • `ConsentManagementEnabled` (available in API version 45.0 and later) • `EnhancedEmailEnabled` • `EventLogWaveIntegEnabled` • `LoginForensicsEnabled` • `NetworksEnabled` (available in API version 40.0 and later) • `NotesReservedPref01` • `OfflineDraftsEnabled` • `PathAssistantsEnabled` • `S1DesktopEnabled` Note: After it is enabled, `S1DesktopEnabled` can't be disabled in any version of the API. • `S1EncryptedStoragePref2` • `S1OfflinePref` • `ScratchOrgManagementPref` `on page 1901` (available in API version 41.0 and later) • `SendThroughGmailPref` • `SocialProfilesEnable` • `Translation` (available in API version 40.0 and later) • `VoiceEnabled` Note: The `VoiceEnabled` preference isn't being moved to another metadata type. If you want to use it in a scratch org in API version 48.0 and later, you can enable it as a scratch org feature.

OrganizationSettingsDetail

Field Name	Field Type	Description
`settingName`	string	The name of the setting. For example, "S1EncryptedStoragePref2."
`settingValue`	boolean	Indicates whether the setting is enabled (`true`) or not (`false`).

Declarative Metadata Sample Definition

The following is an example of a OrgPreferenceSettings component. The example shows only the `preferences` values that are supported but not yet available as fields on another Settings type in API version 47.0.

```
<?xml version="1.0" encoding="UTF-8"?>
<OrgPreferenceSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <preferences>
        <settingName>AnalyticsSharingEnable</settingName>
        <settingValue>true</settingValue>
    </preferences>
    <preferences>
        <settingName>NetworksEnabled</settingName>
        <settingValue>true</settingValue>
    </preferences>
    <preferences>
        <settingName>NotesReservedPref01</settingName>
        <settingValue>false</settingValue>
    </preferences>
    <preferences>
        <settingName>ScratchOrgManagementPref</settingName>
        <settingValue>true</settingValue>
    </preferences>
    <preferences>
        <settingName>VoiceEnabled</settingName>
        <settingValue>false</settingValue>
    </preferences>
</OrgPreferenceSettings>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

OrgSettings

Represents the settings for org-wide functionality that isn't associated with any specific feature.This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

A OrgSettings component file has the suffix `.settings` and is stored in the `settings` directory. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

OrgSettings components are available in API version 46.0 and later.

Before API version 51.0, the fields `enableExtendedMailMerge` and `saveMailMergeDocsAsSalesforceDocs` were found within OrgSettings components. In API version 51.0 and later, those fields are found within MailMergeSettings on page 2021.

Fields

Field Name	Field Type	Descriptions
`enableCustomerSuccessPortal`	boolean	Indicates whether Customer Portal is enabled (`true`) or not (`false`).
`enableManageSelfServiceUsers`	boolean	Indicates whether mass management of self-service users is enabled through the Self-Service Portal (`true`) or not (`false`).
`enableOrgFeedSentimentAnalysis`	boolean	Indicates whether feed sentiment analysis is enabled for the org (`true`) or not (`false`).

Declarative Metadata Sample Definition

The following is an example of a OrgSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<OrgSettings xmlns="http://soap.sforce.com/2006/04/metadata"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <enableCustomerSuccessPortal>false</enableCustomerSuccessPortal>
    <enableMakeDeploymentsMandatory>true</enableMakeDeploymentsMandatory>
    <enableManageSelfServiceUsers>false</enableManageSelfServiceUsers>
    <enableOrgFeedSentimentAnalysis>false</enableOrgFeedSentimentAnalysis>
    <enableRADeploymentAttributeOnly>true</enableRADeploymentAttributeOnly>
    <enableResetDivisionOnLogin xsi:nil="true"/>
</OrgSettings>
```

Example Package Manifest

The following is an example package manifest used to deploy or retrieve the org settings metadata for an organization:

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>Org</members>
        <name>Settings</name>
    </types>
    <version>47.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

PartyDataModelSettings

Represents an organization's party data model settings, including options around the Individual object and consent enablement. This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

PartyDataModelSettings values are stored in the `PartyDataModel.settings` file in the `settings` directory. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

PartyDataModelSettings is available in API version 47.0 and later.

Fields

Field Name	Field Type	Description
`enableAutoSelectIndividualOnMerge`	boolean	Indicates whether the most recently modified data privacy record for the Individual is retained when merging lead, contact, and person accounts (`true`) or users must manually determine which data privacy record to retain during the merge process (`false`). This field has a default value of `false`.
`enableConsentManagement`	boolean	Indicates whether data protection details are available in records (`true`) or not (`false`). This has a default value of `true`. Note: Setting this field to `false` purges all data protection details, such as privacy preferences and stored consent forms.
`enableIndividualAutoCreate`	boolean	Deprecated in API version 48.0 and removed in API version 49.0 and later.

Declarative Metadata Sample Definition

The following is an example of a PartyDataModelSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<PartyDataModelSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <enableAutoSelectIndividualOnMerge>true</enableAutoSelectIndividualOnMerge>
    <enableConsentManagementEnabled>true</enableConsentManagementEnabled>
</PartyDataModelSettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>PartyDataModel</members>
        <name>Settings</name>
    </types>
    <version>47.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

PardotSettings

Represents Marketing Cloud Account Engagement settings in your Salesforce org. Account Engagement, formerly known as Pardot, is a B2B marketing automation solution that helps you create meaningful connections, generate more pipeline, and close more deals. Use these settings to configure how Account Engagement collects and displays data.

This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

This object is stored in a file named `Pardot.Settings` in the `settings` folder of the corresponding package directory. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

PardotSettings is available in API version 47.0 and later.

Special Access Rules

This metadata type is available only to orgs with Account Engagement.

Fields

Field Name	Field Type	Description
`cdpEnablementStatus`	string	The status of the enablement in Data Cloud for the account engagement business unit.
`enableAeDataConnector`	boolean	Enable the Account Engagement Data Cloud Connector for creating Account Engagement Data Streams.

Field Name	Field Type	Description
enableAIOptimizedSendTime	boolean	Enable Einstein Send Time Optimization for sending Account Engagement emails.
enableB2bmaAppEnabled	boolean	Deprecated.
enableEngagementHistoryDashboards	boolean	Enable the Engagement History Dashboard and allow related Account Engagement data to be shared to campaign records in Salesforce by setting this value to `true`. The default value is `false`. If `enableEnagementHistoryDashboards` is disabled after being enabled, the Engagement History Dashboard is removed, but engagement data continues to be retained and updated.
enableEnhancedProspectCustomFieldsSync	boolean	Enable Object Sync to enhance with B2B Marketing Analytics or B2B Marketing Analytics Plus by setting this property to `true`. The default value is `false`. Available in API version 52.0 and later.
enablePardotAppV1Enabled	boolean	Enable the Account Engagement Lightning App by setting this property to `true`. The default value is `false`.
enablePardotEnabled	boolean	Deprecated.
enablePardotObjectSync	boolean	Deprecated.
enableProspectActivityDataset	boolean	Enable the Prospect and Activity Dataset for B2B Marketing Automation apps by setting this property to `true`. When `enableProspectActivityDataset` is `true`, the datasets take some time to populate. Depending on how much data and the type of licenses you have, enabling this preference can impact the account's row limit for Analytics. If `enableProspectActivityDataset` is disabled after being enabled: • The data that makes up the datasets is deleted. • The Prospect and Activity Dataset in existing B2B Marketing Automation apps stops getting updates. • The dataset isn't available to add to new apps. • When apps are reconfigured, the dataset is deleted. Requires that `enableEnagementHistoryDashboards` is set to `true`.
PardotEngageFreqSetting	boolean	Enable Einstein Engagement Frequency for sending Account Engagement emails.

Declarative Metadata Sample Definition

The following is an example of a PardotSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<PardotSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <enablePardotEnabled>true</enablePardotEnabled>
```

```
4     <enablePardotAppV1Enabled>true</enablePardotAppV1Enabled>
5     <enableB2bmaAppEnabled>true</enableB2bmaAppEnabled>
6     <enableEngagementHistoryDashboards>true</enableEngagementHistoryDashboards>
7    <enableEnhancedProspectCustomFieldsSync>true</enableEnhancedProspectCustomFieldsSync>
8     <enablePardotObjectSync>true</enablePardotObjectSync>
9     <enableProspectActivityDataset>true</enableProspectActivityDataset>
10     <enableAIOptimizedSendTime>true</enableAIOptimizedSendTime>
11 </PardotSettings>
```

The following is an example `package.xml` that references the previous definition.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <Package xmlns="http://soap.sforce.com/2006/04/metadata">
3     <types>
4         <members>Pardot</members>
5         <name>Settings</name>
6     </types>
7     <version>47</version>
8 </Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

PardotEinsteinSettings

Represents PardotEinsteinSettings. Use these settings to learn what factors drive your campaign performance, and get the best possible engagement score for your prospects. This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

PardotEinsteinSettings values are stored in the `PardotEinstein.settings` file in the `settings` folder. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

PardotEinsteinSettings is available in API versions 48.0 and later.

Fields

Field Name	Field Type	Description
`enableCampaignInsight`	boolean	Indicates whether Einstein Campaign Insights is enabled (`true`) or not (`false`). Einstein Campaign Insights helps you understand what factors drive campaign performance. The default value is `false`.

Field Name	Field Type	Description
`enableEngagementScore`	boolean	Indicates whether Einstein Behavior Scoring is enabled (`true`) or not (`false`). Einstein Behavior Scoring identifies prospects whose behavior suggests that they are ready to buy, and scores them based on Einstein's engagement model. The default value is `false`.

Declarative Metadata Sample Definition

The following is an example of the PardotEinstein.settings file:

```
<?xml version="1.0" encoding="UTF-8"?>
<PardotEinsteinSettings xmlns="http://soap.sforce.com/2006/04/metadata">
 <enableCampaignInsight>true</enableCampaignInsight>
 <enableEngagementScore>true</enableEngagementScore>
</PardotEinsteinSettings>
```

Example Package Manifest

The following is an example package manifest used to deploy or retrieve the PardotEinstein settings metadata:

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
 <types>
  <members>PardotEinstein</members>
  <name>Settings</name>
 </types>
 <version>29.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

PathAssistantSettings

Represents the Path preference setting. This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

PathAssistantSettings components have the suffix `.settings` and are stored in the `settings` folder.

Version

PathAssistantSettings components are available in API version 34.0 and later.

Fields

Field Name	Field Type	Description
canOverrideAutoPathCollapseWithUserPref	boolean	Keeps a user's path expanded to show guidance and key fields on all their records. A user's path stays expanded until the user collapses it. To use this preference, Path must be enabled. Default value is `false` for all editions. When set to `false`, the user's path is collapsed when the page loads. Available in API version 47.0 and later.
pathAssistantEnabled	boolean	Determines whether the preference is enabled for Path. Default value is `true` for Enterprise Edition and `false` for other editions. Available in API version 35.0 and later.
pathAssistantForOpportunityEnabled	boolean	Determines whether the preference is enabled for Path in Opportunity or not. Available in API version 34.0 and earlier.

Declarative Metadata Sample Definition

The following is an example of a PathAssistantSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<PathAssistantSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <pathAssistantEnabled>true</pathAssistantEnabled>
   <canOverrideAutoPathCollapseWithUserPref>true</canOverrideAutoPathCollapseWithUserPref>
</PathAssistantSettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>PathAssistant</members>
        <name>Settings</name>
    </types>
    <version>API</version>
</Package
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

PaymentsSettings

Represents the Salesforce Payments settings when this feature is enabled for the org.

Parent Type and Manifest Access

This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all the settings metadata types for the org are accessed using the "Settings" name. See Settings for more details.

File Suffix and Directory Location

`PaymentsSettings` values are stored in the `Payments.settings` file in the `settings` folder.

The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

PaymentsSettings is available in API version 57.0 and later.

Special Access Rules

This metadata type is only accessible by developers and customers using Salesforce Payments.

Fields

Field Name	Description
`enablePayments`	**Field Type** boolean **Description** Indicates whether Salesforce Payments is enabled (`true`) or not (`false`) for an org. The default is false.

Declarative Metadata Sample Definition

The following is a sample `payments.settings` metadata file.

```
<?xml version="1.0" encoding="UTF-8"?>
   <PaymentsSettings xmlns="http://soap.sforce.com/2006/04/metadata">
     <enablePayments>true</enablePayments>
   </PaymentsSettings>
```

The following is an example `package.xml` that references the previous definition.

```
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>Payments</members>
        <name>Settings</name>
    </types>
```

```
    <version>57.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

PicklistSettings

Represents an org's picklist settings. These settings control the behavior of a picklist. This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

PicklistSettings values are stored in a single file named `Picklist.settings` in the `settings` directory. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

Picklist settings are available in API version 47.0 and later.

Fields

Field Name	Field Type	Description
`isPicklistApiNameEditDisabled`	boolean	While `true`, users, including admins with Customize Application permission, can't change the API name of a picklist field. Formulas reference a picklist's API name so that the formula continues to work even if the displayed name value changes. Prevent changes to the API name to protect the references to fields in formulas or during integrations, such as during a data import. The default is `false`.

Declarative Metadata Sample Definition

The following is a sample `picklist.settings` metadata file.

```
<?xml version="1.0" encoding="UTF-8"?>
<PicklistSettings xmlns="http://soap.sforce.com/2006/04/metadata">
  <isPicklistApiNameEditDisabled>true</isPicklistApiNameEditDisabled>
</PicklistSettings>
```

The following is an example `package.xml` manifest that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
 <types>
  <members>Picklist</members>
  <name>Settings</name>
 </types>
 <version>47.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

PlatformEncryptionSettings

Represents an org's Platform Encryption settings, such as settings for available encryption schemes, permissions, encryption policy access, and which fields can be encrypted. This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

PlatformEncryptionSettings values are stored in the `PlatformEncryption.settings` file in the `settings` folder. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

PlatformEncryptionSettings is available in API versions 47.0 and later.

Special Access Rules

To enable and disable PlatformEncryptionSettings attributes, you need the Customize Application permission. Attributes that allow key management tasks require the Manage Encryption Keys permission. For a complete list of required permissions, read Which User Permissions Does Shield Platform Encryption Require?.

Fields

Field Name	Field Type	Description
`canEncryptManagedPackageFields`	boolean	Indicates whether users can enable encryption on custom fields in installed managed packages (`true`) or not (`false`).
`isUseHighAssuranceKeysRequired`	boolean	This field is for internal use.

Field Name	Field Type	Description
`isMEKForEncryptionRequired`	boolean	Indicates whether encryption policy tasks, such as enabling encryption on fields, also require the Manage Encryption Keys permission (`true`) or not (`false`), in addition to those tasks' baseline permissions.
`enableDeterministEncryption`	boolean	Indicates whether customers apply the deterministic encryption scheme to supported fields (`true`) or not (`false`). The deterministic encryption scheme lets customers filter on encrypted data..
`enableEncryptFieldHistory`	boolean	Indicates whether the background encryption process applies the customer's active key material to field history and feed tracking values (`true`) or not (`false`). The default value is `false`. If `false`, background encryption processes apply active key material to all encrypted data except duplicates of that data stored in field history or feed tracking.
`enableEventBusEncryption`	boolean	This field is for internal use.

Declarative Metadata Sample Definition

The following is an example of the PlatformEncryption.settings file:

```
<?xml version="1.0" encoding="UTF-8"  standalone="yes"?>
<PlatformEncryptionSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <canEncryptManagedPackageFields>true</canEncryptManagedPackageFields>
    <isUseHighAssuranceKeysRequired>true</isUseHighAssuranceKeysRequired>
    <isMEKForEncryptionRequired>true</isMEKForEncryptionRequired>
    <enableDeterministEncryption>true</enableDeterministEncryption>
   <enableEncryptFieldHistory>true</enableEncryptFieldHistory></PlatformEncryptionSettings>
```

Example Package Manifest

The following is an example package manifest used to deploy or retrieve the Platform Encryption settings metadata for an organization:

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>PlatformEncryption</members>
        <name>Settings</name>
    </types>
    <version>47.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

PlatformEventSettings

Represents settings for platform events and change data capture events.

Parent Type and Manifest Access

This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all the settings metadata types for the org are accessed using the "Settings" name. See Settings for more details.

File Suffix and Directory Location

`PlatformEventSettings` values are stored in the `PlatformEvent.settings` file in the `settings` folder. The `.settings` files are different from other named components, because there's only one settings file for each settings component.

Version

PlatformEventSettings components are available in API version 58.0 and later.

Special Access Rules

There are no additional access requirements that are specific to this type.

Fields

Field Name	Description
`enableEnhancedUsageMetrics`	**Field Type** boolean **Description** Enables enhanced usage metrics for queries run against PlatformEventUsageMetric. Enhanced usage metrics provide additional fields for the queries and granular time segments. For more information, see Enhanced Usage Metrics in the *Platform Events Developer Guide*. Default value is `false`.

Declarative Metadata Sample Definition

The following is an example of a PlatformEventSettings component that enables the enhanced usage metrics feature.

```
<?xml version="1.0" encoding="UTF-8"?>
<PlatformEventSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <enableEnhancedUsageMetrics>true</enableEnhancedUsageMetrics>
</PlatformEventSettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
```

```
        <members>PlatformEvent</members>
        <name>Settings</name>
    </types>
    <version>58.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

PredictionBuilderSettings

Represents the settings that determine how a user can interact with Einstein Prediction Builder. This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

PredictionBuilderSettings values are stored in the PredictionBuilder.settings file in the settings directory. The .settings files are different from other named components in that each settings component has only one settings file.

Version

PredictionBuilderSettings components are available in API version 47.0 and later.

Special Access Rules

This type is available only if the CRM Analytics Plus or Einstein Predictions license is enabled in your org.

Fields

Field Name	Field Type	Description
`enablePredictionBuilder`	boolean	Indicates whether Einstein Prediction Builder is enabled (`true`) or not (`false`).
`isPredictionBuilderStarted`	boolean	Indicates whether to display the predictions list view to the user (`true`) or not (`false`).

Declarative Metadata Sample Definition

This is a sample Prediction Builder settings file.

```
<?xml version="1.0" encoding="UTF-8"?>
<PredictionBuilderSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <isPredictionBuilderStarted>false</isPredictionBuilderStarted>
```

```
    <enablePredictionBuilder>false</enablePredictionBuilder>
</PredictionBuilderSettings>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

PrivacySettings

Represents an organization's settings for data privacy and consent management. This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

PrivacySettings values are stored in the `Privacy.settings` file in the `settings` directory. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

PrivacySettings components are available in API version 47.0 and later.

Special Access Rules

To use PrivacySettings, you need the Customize Application or Modify Data Classification user permission.

Fields

Field Name	Field Type	Description
`enableConsentAuditTrail`	boolean	Reserved for future use.
`enableConsentEventStream`	boolean	Allows orgs to stream consent changes to the party data model via platform events. This field has a default value of `false`. Available in API version 47.0 and later.
`enableDefaultMetadataValues`	boolean	Indicates whether a default data sensitivity value is applied to all contacts, leads, person accounts, and users (`true`) or not (`false`). This field has a default value of `false`. Available in API version 47.0 and later.
`useUmaDefaultConsentRecs`	boolean	Indicates whether a Preference Manager setup in Privacy Center uses default Marketing Cloud consent parameters and features. This field has a default value of `false`. Available in API version 58.0 and later.

Declarative Metadata Sample Definition

The following is an example of a PrivacySettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<PrivacySettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <enableDefaultMetadataValues>false</enableDefaultMetadataValues>
</PrivacySettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>Privacy</members>
        <name>Settings</name>
    </types>
    <version>47.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ProcessFlowMigration

Represents a process's migrated criteria and the resulting migrated flow.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

Version

ProcessFlowMigration components are available in API version 58.0 and later.

Special Access Rules

Fields

Field Name	Description
`destinationFlowDefinition`	**Field Type** string

Field Name	Description
	Description Required. The ID of the resulting migrated flow.
`destinationFlowVersion`	**Field Type** string **Description** Required. The version ID of the migrated flow.
`developerName`	**Field Type** string **Description** Required. The unique name of the object in the API. This name can contain only underscores and alphanumeric characters, and must be unique in your org. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
`masterLabel`	**Field Type** string **Description** Required. The label for the ProcessFlowMigration.
`migratedCriteriaLabel`	**Field Type** string **Description** The label of the criteria that was migrated.
`migratedCriteriaName`	**Field Type** string **Description** The name of the criteria that was migrated.
`processVersion`	**Field Type** string **Description** Required. The version ID of the originating process.

Declarative Metadata Sample Definition

The following is an example of a ProcessFlowMigration component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ProcessFlowMigration xmlns="http://soap.sforce.com/2006/04/metadata"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
```

```
    <destinationFlowDefinition>Migration_1</destinationFlowDefinition>
    <destinationFlowVersion>Migration_1-1</destinationFlowVersion>
    <developerName>Migration</developerName>
    <masterLabel>Migration_1</masterLabel>
    <migratedCriteriaLabel>myCriteria_1</migratedCriteriaLabel>
    <migratedCriteriaName>myDecision</migratedCriteriaName>
    <processVersion>Migration-1</processVersion>
</ProcessFlowMigration>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>ProcessFlowMigration</name>
    </types>
    <types>
        <members>*</members>
        <name>Flow</name>
    </types>
    <version>58.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ProductSettings

Represents organization preferences for quantity schedules, revenue schedules, and active flag interaction with prices. This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

ProductSettings values are stored in a single file named `Product.settings` in the `settings` directory of the corresponding package directory. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

ProductSettings is available in API version 28.0 and later.

Fields

Field Name	Field Type	Description
enableCascadeActivateToRelatedPrices	boolean	When changing active flag on a product record, automatically updates active flag on related prices.

Field Name	Field Type	Description
enableMySettings	boolean	Moves users' personal settings pages from Setup to a separate My Settings pane (true) or not (false). When set to (true), Salesforce makes a reorganized Setup pane accessible to admins via one click in the header. This setting affects all users in your organization. The default is true. Available in API version 47.0 and later.
enableQuantitySchedule	boolean	Enables quantity schedules for products.
enableRevenueSchedule	boolean	Enables revenue schedules for products.

Declarative Metadata Sample Definition

The following is an example of the package file.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>Product</members>
        <name>Settings</name>
    </types>
    <version>28.0</version>
</Package>
```

The package file references the following Product.settings file.

```
<?xml version="1.0" encoding="UTF-8"?>
<ProductSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <enableCascadeActivateToRelatedPrices>true</enableCascadeActivateToRelatedPrices>
    <enableQuantitySchedule>false</enableQuantitySchedule>
    <enableRevenueSchedule>false</enableRevenueSchedule>
</ProductSettings>
```

Wildcard Support in the Manifest File

The wildcard character * (asterisk) in the package.xml manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

QuoteSettings

Represents an org's quotes settings, such as enabling quotes or creating quotes without an associated opportunity. This type extends the Metadata metadata type and inherits its fullName field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

QuoteSettings values are stored in a single file named Quote.settings in the settings directory of the corresponding package directory. The .settings files are different from other named components because there's only one settings file for each settings component.

Version

QuoteSettings is available in API version 28.0 and later.

Fields

Field Name	Field Type	Description
`enableQuote`	boolean	When set to `true`, users can access Quotes.
`enableQuotesWithoutOppEnabled`	boolean	When set to `true`, users can create quotes independently of an opportunity. For example, a user can create a quote for budgeting purposes, before creating the Opportunity. Default value is `false`. When set to `false`, users can only create quotes from an Opportunity. Before setting to `false`, delete any quotes that do not have opportunities. Available in API version 47.0 and later.

Declarative Metadata Sample Definition

The following is an example of the package file.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>Quote</members>
        <name>Settings</name>
    </types>
    <version>28.0</version>
</Package>
```

The package file references the following Quote.settings file.

```
<?xml version="1.0" encoding="UTF-8"?>
<QuoteSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <enableQuote>true</enableQuote>
    <enableQuotesWithoutOppEnabled>true</enableQuotesWithoutOppEnabled>
</QuoteSettings>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

RealTimeEventSettings

Represents the list of Real-Time Event entities that you want to enable or disable. This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

Real-Time Event settings are stored in a single file named `RealTimeEvent.settings` in the `settings` directory. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

RealTimeEventSettings is available in API version 50.0 and later.

Fields

Field Name	Field Type	Description
`realTimeEvents`	RealTimeEvent[]	Represents the list of Real-Time Event entities that you want to enable or disable.

RealTimeEvent

Represents one of the Real-Time Event entities that you want to enable or disable.

Field Name	Field Type	Description
`entityName`	string	The storage or streaming entity name that you want to modify. For example: ApiEvent or ApiEventStream.
`isEnabled`	boolean	Indicates whether you want the storage or streaming capability to be enabled (`true`) or disabled (`false`).

Declarative Metadata Sample Definition

The following is an example `RealTimeEvent.settings` metadata file:

```
<?xml version="1.0" encoding="UTF-8"?>
<RealTimeEventSettings xmlns="http://soap.sforce.com/2006/04/metadata"/>
  <realTimeEvents>
    <entityName>ApiEventStream</entityName>
    <isEnabled>true</isEnabled>
  </realTimeEvents>
  <realTimeEvents>
    <entityName>ApiEvent</entityName>
    <isEnabled>true</isEnabled>
  </realTimeEvents>
</RealTimeEventSettings>
```

The following is an example `package.xml` manifest that references the RealTimeEventSettings definitions:

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
```

```
    <types>
        <members>RealTimeEvent</members>
        <name>Settings</name>
    </types>
    <version>51.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

RecordPageSettings

Represents an org's record page settings. This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

Declarative Metadata File Suffix and Directory Location

RecordPageSettings values are stored in a single file named `RecordPage.settings` in the `settings` directory. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

Record page settings are available in API version 47.0 and later.

Fields

Field	Field Type	Description
`enableActivityRelatedList`	boolean	Indicates whether the default activities view is related lists (`true`) or activity timeline (`false`).
`enableDynamicForms`	boolean	Indicates whether Dynamic Forms is enabled for the org. Removed in API version 50.0 and later.
`enableFullRecordView`	boolean	Indicates whether the default record page view is full view (`true`) or grouped view (`false`).

Declarative Metadata Sample Definition

This is a sample `recordpage.settings` metadata file.

```
<?xml version="1.0" encoding="UTF-8"?>
<RecordPageSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <enableDynamicForms>true</enableDynamicForms>
    <enableActivityRelatedList>true</enableActivityRelatedList>
    <enableFullRecordView>true</enableFullRecordView>
</RecordPageSettings>
```

Example Package Manifest

The following is an example package manifest used to deploy or retrieve the Record Page settings metadata for an organization

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>RecordPage</members>
        <name>Settings</name>
    </types>
    <version>47.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SEE ALSO:

Settings

RetailExecutionSettings

Represents settings to manage your inventory, promotions, planograms, and in-store activities.

This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for more details.

File Suffix and Directory Location

RetailExecutionSettings are stored in a single file named `RetailExecution.settings` in the `settings` directory.

Version

RetailExecutionSettings are available in API version 47.0 and later.

Fields

Field Name	Field Type	Description
`enableProductHierarchy`	boolean	Indicates whether Product Hierarchy is enabled for your org (`true`) or not `false`). This field is available in API version 53.0 and later.
`enableRetailExecution`	boolean	Indicates whether Retail Execution is enabled for your org (`true`) or not (`false`). The default value is `false`.
`enableVisitSharing`	boolean	Indicates whether Visit Share is enabled for your org (`true`) or not (`false`). The default value is `false`. This field is available in API version 55.0 and later.

Declarative Metadata Sample Definition

The following is an example of a RetailExecutionSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<RetailExecutionSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <enableRetailExecution>true</enableRetailExecution>
    <enableProductHierarchy>true</enableProductHierarchy>
    <enableVisitSharing>false</enableVisitSharing>
</RetailExecutionSettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>RetailExecution</members>
        <name>Settings</name>
    </types>
    <version>55.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SalesAgreementSettings

Represents settings that control the display of agreement terms metrics in sales agreements and the calculation of the actual quantity of products in sales agreements. These settings also control the approval of sales agreements.

This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for more details.

File Suffix and Directory Location

SalesAgreementSettings values are stored in the `SalesAgreementSettings.salesAgreementSetting` file in the `salesAgreementSettings` directory.

Version

SalesAgreementSettings components are available in API version 47.0 and later.

Fields

Field Name	Field Type	Description
`actualsCalculationMode`	ActualsCalculationMode (enumeration of type string)	Required. Source from which the actual ordered quantity of a product in a sales agreement is calculated. Valid values are: • `Manual: Default` • `Orders` • `OrdersThroughContracts`
`displayGroups`	AdvAcctFrcstDisplayGroup	Represents information about the groups for the advanced account forecast set measures or dimensions. Available in API version 56.0 and later.
`displayedAgreementTermsMetrics`	string	Required. Metrics that are selected for display in the sales agreement terms in the specified sequence. There can be a maximum of 10 comma-separated metric names in this list.
`isOnlyApprovalProcessUsed`	boolean	Required. Indicates whether both self-approval and approval of sales agreements are allowed (`true`) or only approval of sales agreements is allowed `false`) through the Approval Flow.
`measureDefinitions`	AdvAcctForecastMeasureDef on page 2085	Represents information about the measures to be displayed in the advanced account forecasts grid for the forecast set. Available in API version 56.0 and later.
`objectMapping`	ObjectMapping on page 2086	Foreign key to ObjectMapping on page 2086 that maps fields from the input object of SalesAgreementSettings to fields in the output object of SalesAgreementSettings.
`primaryNotifEmailAddress`	string	The email address to which notifications are sent.
`renewalPeriodDayCount`	int	The number of days before the end date of a sales agreement from when the agreement can be renewed. Available in API version 50.0 and later.
`secondaryNotifEmailAddress`	string	The second email address to which notifications are sent.

AdvAcctFrcstDisplayGroup

Represents information about the groups for the advanced account forecast set measures or dimensions. Available in API version 56.0 and later.

Field Name	Field Type	Description
advAcctFrcstDisplayGroupName	string	Required. Name of the advanced account forecast display group.
displayGroupItems	AdvAcctFrcstDplyGroupItem on page 2085	Represents information about the items associated with a display group for an advanced account forecast set.
displayGroupType	AdvAcctFrcstDplyGroupType (enumeration of type string)	Category for the display group. Possible values are: • MEASURE
isDefault	boolean	Indicates whether the display group is the default group (true) or not (false). The default value is false.
userProfileName	string	Profile for which the display group is applicable.

AdvAcctFrcstDplyGroupItem

Represents information about the items associated with a display group for an advanced account forecast set. Available in API version 56.0 and later.

Field Name	Field Type	Description
advAcctFrcstDplyGroupItemName	string	Required. Name of the advanced account forecast display group item.
displayOrder	string	Required. Display order of the display group item.
measureReferenceName	string	Name of the measure associated with the display group item.

AdvAcctForecastMeasureDef

Represents information about the measures to be displayed in the advanced account forecasts grid for the forecast set. Available in API version 56.0 and later.

Field Name	Field Type	Description
advAcctForecastMeasureDefName	string	Required. Name for the measure.
aggregationType	AdvAcctAggregationType (enumeration of type string)	Required. Type of aggregation used for calculating advanced account forecast values. Possible values are: • AVERAGE • MAXIMUM • MINIMUM • SUM

Field Name	Field Type	Description
computationMethod	AdvancedForecastComputationMethod (enumeration of type string)	Required. Method used for calculating advanced account forecast values. Possible values are: • CUSTOM • DATA_PROCESSING_ENGINE_DEFINITION • FORMULA
forecastDataMeasureName	string	Required. Field of the facts object used for this measure.
forecastMeasureName	string	Required. Name for the measure to show on UI.
forecastMeasureType	AdvancedForecastMeasureType (enumeration of type string)	Required. Measure type used for the generated advanced forecast values. Possible values are: • QUANTITY • REVENUE
isAdjustmentTracked	boolean	Indicates whether the adjustments made to the advanced account forecast values for this metric are tracked (true) or not (false). The default value is false.

ObjectMapping

Represents a map of fields in the input object of SalesAgreementSettings to fields in the output object of SalesAgreementSettings. The input object is SalesAgreementProductSchedule. The output object is SalesAgreementProduct.

Field Name	Field Type	Description
inputObject	string	Required. The input object for the SalesAgreementSettings. SalesAgreementProductSchedule is the input object for the SalesAgreementSettings.
mappingFields	ObjectMappingField on page 2086	The mapping of source object fields to target object fields for SalesAgreementSettings.
outputObject	string	Required. The output object for the SalesAgreementSettings. SalesAgreementProduct is the output object for the SalesAgreementSettings.

ObjectMappingField

A field name in the SalesAgreementProductSchedule object and the corresponding field name in the SalesAgreementProduct object.

For example, you can create a field named Revenue on the SalesAgreementProductSchedule object and a field named Total Revenue on the SalesAgreementProduct object. To view these field values in the agreement terms of a sales agreement, select the input object as SalesAgreementProductSchedule and the output object as SalesAgreementProduct. In this case, the input field is Revenue and the output field is Total Revenue.

Field Name	Field Type	Description
`inputField`	string	Required. Field in the object specified by the `inputObject` field in ObjectMapping on page 2086. This field is mapped to the field in `outputField`, which is a field in the object specified by the `outputObject` field in ObjectMapping on page 2086.
`outputField`	string	Required. Field in the object specified by the `outputObject` field in ObjectMapping on page 2086. This field is mapped to the field name in `inputField`, which is a field in the object specified by the `inputObject` field in ObjectMapping on page 2086.

Declarative Metadata Sample Definition

The following is an example of SalesAgreementSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<SalesAgreementSettings
 xmlns="http://soap.sforce.com/2006/04/metadata"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <actualsCalculationMode>Orders</actualsCalculationMode>
 <displayGroups>
  <advAcctFrcstDisplayGroupName>Test Measure Group</advAcctFrcstDisplayGroupName>
  <displayGroupItems>
   <advAcctFrcstDplyGroupItemName>PlannedQuantity</advAcctFrcstDplyGroupItemName>
   <displayOrder>1</displayOrder>
   <measureReferenceName>PlannedQuantity</measureReferenceName>
  </displayGroupItems>
  <displayGroupType>MEASURE</displayGroupType>
  <isDefault>false</isDefault>
  <userProfileName xsi:nil="true"/>
 </displayGroups>

<displayedAgreementTermsMetrics>PlannedQuantity,ActualQuantity,SalesPrice,DiscountPercentage,DerivedPlannedAmount</displayedAgreementTermsMetrics>

 <isOnlyApprovalProcessUsed>false</isOnlyApprovalProcessUsed>
 <measureDefinitions>
  <advAcctForecastMeasureDefName>PlannedQuantity</advAcctForecastMeasureDefName>
  <aggregationType>MINIMUM</aggregationType>
  <computationMethod>DATA_PROCESSING_ENGINE_DEFINITION</computationMethod>
  <forecastDataMeasureName>PlannedQuantity</forecastDataMeasureName>
  <forecastMeasureName>PlannedQuantity</forecastMeasureName>
  <forecastMeasureType>QUANTITY</forecastMeasureType>
  <isAdjustmentTracked>true</isAdjustmentTracked>
 </measureDefinitions>
 <secondaryNotifEmailAddress>abc@salesforce.com</secondaryNotifEmailAddress>
 <primaryNotifEmailAddress>abc@salesforce.com</primaryNotifEmailAddress>
 <renewalPeriodDayCount>50</renewalPeriodDayCount>
 <objectMapping>
  <inputObject>SalesAgreementProductSchedule</inputObject>
  <mappingFields>
   <inputField>SAPS1__c</inputField>
   <outputField>SAP1__c</outputField>
```

```
  </mappingFields>
  <outputObject>SalesAgreementProduct</outputObject>
 </objectMapping>
</SalesAgreementSettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
     <types>
         <members>SalesAgreementProduct.SAP1__c</members>
         <members>SalesAgreementProductSchedule.SAPS1__c</members>
         <name>CustomField</name>
     </types>
     <types>
         <members>*</members>
         <name>SalesAgreementSettings</name>
     </types>
     <version>64.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SandboxSettings

Represents Sandbox settings. This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

SandboxSettings values are stored in the `Sandbox.settings` file in the `settings` folder. The `.settings` files are different from other named components because there is only one settings file for each settings component.

Version

SandboxSettings are available in API version 56.0 and later.

Fields

Field Name	Field Type	Description
`disableSandboxExpirationEmails`	boolean	Indicates whether to disable sandbox expiration email notifications for the source (production) org: `true` or `false`. When disabled in the source (production) org, users no longer receive email notifications for impending deletions of sandboxes that have been inactive for 180 days or longer.

Declarative Metadata Sample Definition

The following is an example of a SandboxSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<SandboxSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <disableSandboxExpirationEmails>true</disableSandboxExpirationEmails>
</SandboxSettings>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SchemaSettings

Represents an org's schema settings, which manage the availability of custom settings and custom metadata type values. This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

SchemaSettings values are stored in the `Schema.settings` file in the `settings` directory. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

SchemaSettings is available in API version 47.0 and later.

Fields

Field Name	Field Type	Description
`enableAdvancedCMTSecurity`	boolean	Indicates whether custom metadata type values are available only to Apex, flow, and formula operations (`true`) or exposed in other contexts such as through the Enterprise WSDL or SOAP API (`false`). This field has a default value of `false`.
`enableAdvancedCSSecurity`	boolean	Indicates whether custom settings type values are available only to Apex, flow, and formula operations (`true`) or exposed in other contexts such as through the Enterprise WSDL or SOAP API (`false`). This field has a default value of `false`.
`enableListCustomSettingCreation`	boolean	Indicates whether you can create custom settings when using application-level data definitions (`true`) or not (`false`). This field has a default value of `false`.

Field Name	Field Type	Description
enableSOSLOnCustomSettings	boolean	Indicates whether custom settings values are returned in Salesforce Object Search language (SOSL) queries (true) or not (false). This field has a default value of false.

Declarative Metadata Sample Definition

The following is an example of a SchemaSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<SchemaSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <enableAdvancedCMTSecurity>true</enableAdvancedCMTSecurity>
    <enableAdvancedCSSecurity>true</enableAdvancedCSSecurity>
    <enableListCustomSettingCreation>false</enableListCustomSettingCreation>
    <enableSOSLOnCustomSettings>true</enableSOSLOnCustomSettings>
</SchemaSettings>
```

The following is an example package.xml that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>Schema</members>
        <name>Settings</name>
    </types>
    <version>47.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character * (asterisk) in the package.xml manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SearchSettings

Represents an org's search settings.

This type extends the Metadata metadata type and inherits its fullName field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for more details.

File Suffix and Directory Location

SearchSettings values are stored in a single file named Search.settings in the settings folder. The .settings files are different from other named components because there's only one settings file for each settings component.

Version

SearchSettings is available in API version 37.0 and later.

Fields

Field Name	Field Type	Description
documentContentSearchEnabled	boolean	Indicates whether a full-text document search is performed.
enableAdvancedSearchInAlohaSidebar	boolean	Indicates whether advanced search is available in the search sidebar (true) or not (false). Available in Salesforce Classic only. Available in API version 46.0 and later.
enableEinsteinSearchAssistantDialog	boolean	Indicates whether the Einstein search experience is enabled (true) or not (false). Available in API version 50.0 and later.
enableEinsteinSearchEs4kPilot	boolean	Indicates whether Einstein Search for Knowledge enhancements are enabled (true) or not (false). Available in API version 54.0 and later. This feature became generally available in Winter '23. In API version 56.0 and later, the default value is true.
enableEinsteinSearchNaturalLanguage	boolean	Indicates whether natural language search is enabled (true) or not (false). Available in API version 50.0 and later.
enableEinsteinSearchNLSFilters	boolean	Indicates whether the Natural Language Search Filters (Pilot) feature is enabled (true) or not (false). Available in API version 54.0 and later.
enableEinsteinSearchPersonalization	boolean	Indicates whether search personalization is enabled (true) or not (false). Available in Lightning Experience only. Available in API version 47.0 and later.
enablePersonalTagging	boolean	Indicates whether users are allowed to group records from various objects by a common theme (true) or not (false). Personal tags are visible to the user only. Available in Salesforce Classic only. Available in API version 48.0 and later.
enablePublicTagging	boolean	Indicates whether users are allowed to group records from various objects by a common theme (true) or not (false). Personal tags are visible to all users. Available in Salesforce Classic only. Available in API version 48.0 and later.
enableSalesforceGeneratedSynonyms	boolean	Indicates whether search synonyms are enabled (true) or not (false). Available in API version 47.0 and later.
enableSearchTermHistory	boolean	Indicates whether users are allowed to group records from various objects by a common theme (true) or not (false). Public tags are visible to everyone in the organization. Available in Salesforce Classic only. Available in API version 48.0 and later.
enableSetupSearch	boolean	Indicates whether the search box in the Setup sidebar returns matching custom fields, custom objects, and other supported setup items when you press Enter (true) or not (false). The default is true in Developer, Performance, Professional,

Field Name	Field Type	Description
		Enterprise, and Unlimited editions, and `false` in all other editions. Available in API version 47.0 and later.
`enableSuggestArticlesLinksOnly`	boolean	Indicates whether links are provided to knowledge articles from Cases similar to the current Case (`true`) or not (`false`). Available in API version 48.0 and later.
`enableUseDefaultSearchEntity`	boolean	Indicates whether to use the admin-specified default entity in sidebar search (`true`) or not (`false`). Available in Salesforce Classic only. Available in API version 48.0 and later.
`optimizeSearchForCJKEnabled`	boolean	Required. Indicates whether the search is optimized for the Japanese, Chinese, and Korean languages (`true`) or not (`false`). This setting affects sidebar search and the account search for **Find Duplicates** on a lead record in sidebar search and global search. Enable this option if users are searching mostly in Japanese, Chinese, or Korean, and if the text in searchable fields is mostly in those languages.
`recentlyViewedUsersForBlankLookupEnabled`	boolean	Required. Indicates whether the list of records that are returned from a user autocomplete lookup and from a blank user lookup is taken from the user's recently viewed user records (`true`). Otherwise this setting is `false` if the lookup shows a list of recently accessed user records from across your org (`false`). Only applies to User object blank lookup searches.
`searchSettingsByObject`	SearchSettingsByObject	Required. Represents a list of search settings for each object.
`sidebarAutoCompleteEnabled`	boolean	Required. Indicates whether autocomplete is enabled for sidebar search (`true`) or not (`false`). Autocomplete is when users start typing search terms and sidebar search displays a matching list of recently viewed records.
`sidebarDropDownListEnabled`	boolean	Required. Indicates whether a dropdown list appears in the sidebar search section (`true`) or not (`false`). From this list, users can select to search within tags, within a specific object, or across all objects.
`sidebarLimitToItemsIOwnCheckboxEnabled`	boolean	Required. Indicates whether the **Limit to Items I Own** checkbox appears (`true`) or not (`false`). The checkbox allows your users to include only records for which they are the record owner when entering search queries in the sidebar.
`singleSearchResultShortcutEnabled`	boolean	Required. Indicates whether a shortcut is enabled (`true`) or not (`false`). With the shortcut, users skip the search results page and go directly to the record's detail page when their search returns only a single item. This setting doesn't apply to tags, case comments (in advanced search), and global search.
`spellCorrectKnowledgeSearchEnabled`	boolean	Required. Indicates whether spell check is enabled for Knowledge search (`true`) or not (`false`).

SearchSettingsByObject

Field Name	Field Type	Description
`searchSettingsByObject`	ObjectSearchSetting	Contains a list of search settings for each object.

ObjectSearchSetting

A list of search settings for each object.

Field Name	Field Type	Description
`enhancedLookupEnabled`	boolean	Required. Indicates whether enhanced lookups is enabled for the object (`true`) or not (`false`).
`lookupAutoCompleteEnabled`	boolean	Required. Indicates whether autocomplete is enabled for lookup search (`true`) or not (`false`). Autocomplete is when users edit the lookup field inline by choosing an autosuggestion.
`name`	string	Required. The entity name of the object being configured.
`resultsPerPageCount`	int	Required. The number of search results per page.

Declarative Metadata Sample Definition

The following is an example of the `Search.settings` file.

```
<?xml version="1.0" encoding="UTF-8"?>
    <SearchSettings xmlns="http://soap.sforce.com/2006/04/metadata">
        <enableSetupSearch>false</enableSetupSearch>
        <enableAdvancedSearchInAlohaSidebar>false</enableAdvancedSearchInAlohaSidebar>
        <enableQuerySuggestionPigOn>false</enableQuerySuggestionPigOn>
        <enableSalesforceGeneratedSynonyms>false</enableSalesforceGeneratedSynonyms>
        <enableSearchTermHistory>false</enableSearchTermHistory>
        <enablePublicTagging>false</enablePublicTagging>
        <enablePersonalTagging>false</enablePersonalTagging>
        <enableSuggestArticlesLinksOnly>false</enableSuggestArticlesLinksOnly>
        <enableUseDefaultSearchEntity>false</enableUseDefaultSearchEntity>
            <documentContentSearchEnabled>true</documentContentSearchEnabled>
            <optimizeSearchForCJKEnabled>true</optimizeSearchForCJKEnabled>

<recentlyViewedUsersForBlankLookupEnabled>true</recentlyViewedUsersForBlankLookupEnabled>

            <searchSettingsByObject>
            <searchSettingsByObject>
                <enhancedLookupEnabled>false</enhancedLookupEnabled>
                <lookupAutoCompleteEnabled>false</lookupAutoCompleteEnabled>
                <name>Account</name>
                <resultsPerPageCount>25</resultsPerPageCount>
            </searchSettingsByObject>
            <searchSettingsByObject>
```

```
                <enhancedLookupEnabled>false</enhancedLookupEnabled>
                <lookupAutoCompleteEnabled>false</lookupAutoCompleteEnabled>
                <name>Activity</name>
                <resultsPerPageCount>25</resultsPerPageCount>
            </searchSettingsByObject>
            <searchSettingsByObject>
                <enhancedLookupEnabled>false</enhancedLookupEnabled>
                <lookupAutoCompleteEnabled>false</lookupAutoCompleteEnabled>
                <name>Asset</name>
                <resultsPerPageCount>25</resultsPerPageCount>
            </searchSettingsByObject>
            </searchSettingsByObject>
            <sidebarAutoCompleteEnabled>true</sidebarAutoCompleteEnabled>
            <sidebarDropDownListEnabled>true</sidebarDropDownListEnabled>

<sidebarLimitToItemsIOwnCheckboxEnabled>true</sidebarLimitToItemsIOwnCheckboxEnabled>
            <singleSearchResultShortcutEnabled>true</singleSearchResultShortcutEnabled>
            <spellCorrectKnowledgeSearchEnabled>true</spellCorrectKnowledgeSearchEnabled>

        <enableEinsteinSearchPersonalization>true</enableEinsteinSearchPersonalization>
    </SearchSettings>
```

Example Package Manifest

The following is an example package manifest used to deploy or retrieve the Search settings metadata for an organization.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>Search</members>
        <name>Settings</name>
    </types>
    <version>37.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SecuritySettings

Represents an org's security settings. For example, settings define trusted IP ranges for network access, password and login requirements, session expiration, and single sign-on settings.

This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for more details.

File Suffix and Directory Location

SecuritySettings values are stored in a single file named `Security.settings` in the `settings` directory. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

Security settings are available in API version 27.0 and later. API versions 26 and earlier are no longer available.

Fields

Field Name	Field Type	Description
`canUsersGrantLoginAccess`	boolean	If `true`, users can grant login access to Support. If `false`, only an admin can grant login access. Users can't grant login access to managed packages that are licensed to your entire Salesforce org. Only admins with the Manage Users permission enabled on their profiles can grant access to these publishers. Also, some managed packages don't have login access. If a package isn't listed on the Login Access Policies page, login access isn't available for that package.
`enableAdminLoginAsAnyUser`	boolean	If `true`, the **Administrators Can Log in as Any User** field is enabled. The default isn't enabled (`false`).
`enableAuditFieldsInactiveOwner`	boolean	If `true`, this setting enables audit fields and updating the owner for records that are owned by inactive users. The default value is `false`. This field is available in API version 47.0 and later.
`enableAuraSecureEvalPref`	boolean	If `true`, this setting prevents the creation of function expressions in dynamically created Aura components. The default is `false`. This field is available in API version 47.0 and later.
`enableCoepHeader`	boolean	Indicates whether the Cross-Origin Embedder Policy (COEP) response header is applied to this org's custom Visualforce pages (`true`) or not (`false`). If `true`, externally sourced embedded content loads only when the external origin allows it via CORS or CORP. The default value is `false`. Available in API version 55.0 and later.
`enableCoopHeader`	boolean	Indicates whether the Cross-Origin Opener Policy (COOP) response header is applied to this org's custom Visualforce pages (`true`) or not (`false`). If `true`, each custom Visualforce page opens in a new browsing context group. The default value is `false`. Available in API version 55.0 and later.

Field Name	Field Type	Description
`enableCrossOrgRedirects`	boolean	Indicates whether redirections to other Salesforce orgs are allowed (`true`) or blocked (`false`). In Summer '24 and later, this field is always `false`. This setting applies to user redirections to another Salesforce org via a direct link, a post-action URL, or a post-login URL in Salesforce. An example of a direct link with a redirection is `<a href="/?startURL=targetUrl">linkText">linkText</a>`. Post-action URLs and post-login URLs use a protected URL redirect parameter, such as `retURL`, `startURL`, `saveURL`, `cancelURL`, and `targetURL`. Subsequent redirections can't be verified because they occur outside Salesforce. To allow cross-org redirections, add the URLs for the Salesforce orgs that you own to RedirectWhitelistUrl. Available in API version 59.0 to 60.0.
`enablePermissionsPolicy`	boolean	Indicates whether the pages that Salesforce serves for this org include the `Permissions-Policy` HTTP header. This HTTP header controls access to browser features such as cameras and microphones. When this field is `false`, access to the browser features is always permitted. The default value is `false`. Available in API version 59.0 and later.
`enableRequireHttpsConnection`	boolean	Deprecated in API version 47.0 and later.
`grantCameraAccess`	PermissionsPolicy (enumeration)	When `enablePermissionsPolicy` is `true`, indicates when apps and websites loaded from Salesforce can access the user's camera. Possible values are: • `Always`—All apps and websites loaded from Salesforce can access the user's camera. • `Never`—No apps or websites, including scripts from Salesforce domains, can access the user's camera. • `TrustedUrls`—Only CspTrustedSite entries with `canAccessCamera` set to `true` can access the user's camera. If `enablePermissionsPolicy` is `false`, this field has no effect. This field is available in API version 59.0 and later.
`grantMicrophoneAccess`	Permissions Policy (enumeration)	When `enablePermissionsPolicy` is `true`, indicates when apps and websites loaded from Salesforce can access the user's microphone. Possible values are:

Field Name	Field Type	Description
		• `Always`—All apps and websites loaded from Salesforce can access the user's microphone. • `Never`—No apps or websites, including scripts from Salesforce domains, can access the user's microphone. • `TrustedUrls`—Only CspTrustedSite entries with `canAccessMicrophone` set to `true` can access the user's microphone. If `enablePermissionsPolicy` is `false`, this field has no effect. This field is available in API version 59.0 and later.
`isTLSv12Required`	boolean	Indicates whether connections to or from your Salesforce org must use TLS 1.2 or higher (`true`) or not (`false`). This field has a default value of `false`. Removed in API version 51.0 and later.
`isTLSv12RequiredCommunities`	boolean	Indicates whether connections with your Salesforce sites and portals or Experience Cloud sites must use TLS 1.2 or higher (`true`) or not (`false`). This field has a default value of `false`. Removed in API version 51.0 and later.
`networkAccess`	NetworkAccess	The trusted IP address ranges from which users can always log in without requiring computer activation.
`passwordPolicies`	PasswordPolicies	The requirements for passwords and logins, and assistance with retrieving forgotten passwords.
`sendCspForUncommonClients`	boolean	In rare cases, Salesforce can't identify whether the requesting app or specialized browser supports the `Content-Security-Policy: frame-ancestors` HTTP header directive. In those cases, this field indicates whether that directive is included in (`true`) or omitted from (`false`) the HTTP response header for pages that Salesforce serves for this org. The default value is `false`. When `sendCspForUncommonClients` is `true`, users who access Salesforce via an app or browser that doesn't support the `Content-Security-Policy: frame-ancestors` HTTP header directive can experience errors if that lack of support is unclear. This field is available in API version 59.0 and later.
`sessionSettings`	SessionSettings	The settings for session expiration and security.
`singleSignOnSettings`	SingleSignOnSettings	The settings for single sign-on (SSO).

NetworkAccess

Represents your org's trusted IP address ranges for network access.

Field	Field Type	Description
`ipRanges`	IpRange[]	The trusted IP address ranges from which users can always log in without requiring computer activation. To add an IP range, deploy all existing IP ranges, including the one you want to add. Otherwise, the existing IP ranges are replaced with the ones you deploy. To remove all the IP ranges, leave the networkAccess field blank (<networkAccess></networkAccess>).

IpRange

Defines a range of trusted IP addresses for network access.

Field	Field Type	Description
`description`	string	The description of the trusted IP range. Use this field to identify the range, such as which corporate network corresponds to this range. Available in API version 34.0 and later.
`end`	string	The IP address that defines the high end of a range of trusted addresses.
`start`	string	The IP address that defines the low end of a range of trusted addresses.

PasswordPolicies

Represents your org's password and login policies, which show up under **Security Controls | Password Policies**.

Field	Field Type	Description
`apiOnlyUserHomePageURL`	string	The URL to which users with the API Only User permission are redirected instead of the login page.
`complexity`	Complexity (enumeration of type string)	The types of characters that must be used in a user's password. Valid values are: • `NoRestriction`—Has no requirements and is the least secure option. • `AlphaNumeric`—The default setting. Requires at least one alphabetic character and one number. This value is the default value. • `SpecialCharacters`—Requires at least one alphabetic character, one number, and one of the following characters:

Field	Field Type	Description
		`! " # $ % & ' ( ) * + , - . / : ; < = > ? @ [ \ ] ^ _ ` { | } ~`. • `UpperLowerCaseNumeric`—Requires at least one number, one uppercase letter, and one lowercase letter. This value is available in API version 31.0 and later. • `UpperLowerCaseNumericSpecialCharacters`—Requires at least one number, one uppercase letter, one lowercase letter, and one of the following characters: `! " # $ % & ' ( ) * + , - . / : ; < = > ? @ [\ ] ^ _ ` { | } ~`. This value is available in API version 31.0 and later. • `Any3UpperLowerCaseNumericSpecialCharacters`—Requires at least three of the following options: one number, one uppercase letter, one lowercase letter, and one special character (`! " # $ % & ' ( ) * + , - . / : ; < = > ? @ [ \ ] ^ _ ` { | } ~`). This value is available in API version 46.0 and later.
`enableSetPasswordInApi`	boolean	Deprecated in API version 51.0. Removed in API version 52.0.
`expiration`	Expiration (enumeration of type string)	The length of time until a user password expires and must be changed. Valid values are: • `Never` • `ThirtyDays` • `SixtyDays` • `NinetyDays`. This value is the default value. • `SixMonths` • `OneYear`
`historyRestriction`	string	The number of previous passwords saved for users so that they must always reset a new, unique password. Valid values are `0` through `24` passwords remembered. The maximum value of 24 applies to API version 31.0 and later. In earlier versions, the maximum value is 16. The default value is `3`.
`lockoutInterval`	LockoutInterval (enumeration of type string)	The duration of the login lockout. Valid values are: • `FifteenMinutes`. This value is the default value. • `ThirtyMinutes` • `SixtyMinutes` • `Forever` (must be reset by admin)
`maxLoginAttempts`	MaxLoginAttempts (enumeration of type string)	The number of login failures allowed for a user before the user is locked out. Valid values are:

Field	Field Type	Description
		• `NoLimit` • `ThreeAttempts` • `FiveAttempts` • `TenAttempts`. This value is the default value.
`minimumPasswordLength`	string	The minimum number of characters required for a password. The number can contain from 5 to 50 characters (default is 8). Available in API version 35.0 and later. Before API version 35.0, specify minimum password length with the enumeration `minPasswordLength`, with valid values `FiveCharacters`, `EightCharacters` (default), `TenCharacters`, `TwelveCharacters` (API version 31.0 and later), and `FifteenCharacters` (API version 34.0 and later).
`minimumPasswordLifetime`	boolean	If **Require a minimum 1 day password lifetime** is enabled (`true`), passwords can't be changed more than one time during a 24-hour period. The default is `false`. Available in API version 31.0 and later.
`obscureSecretAnswer`	boolean	If enabled (`true`), hide answers to security questions as the user types. The default is `false`. If your org uses the Microsoft Input Method Editor (IME) with the input mode set to Hiragana, when you type ASCII characters, they're converted in to Japanese characters in normal text fields. However, the IME doesn't work properly in fields with obscured text. If your org's users can't properly enter their passwords or other values after enabling this feature, disable the feature.
`passwordAssistanceMessage`	string	The text that appears in the Account Lockout email and at the bottom of the Confirm Identity screen for users resetting their passwords.
`passwordAssistanceURL`	string	The URL that users can click to retrieve forgotten passwords.
`questionRestriction`	QuestionRestriction (enumeration of type string)	The restriction on whether the answer to the password hint question can contain the password itself. Valid values are: • `None` • `DoesNotContainPassword`. This value is the default value.

SessionSettings

Represents your org's session expiration and security settings.

Field	Field Type	Description
allowUserAuthenticationByCertificate	boolean	If enabled (true), users can authenticate with a PEM-encoded X.509 digital certificate. Not enabled by default. Available in API version 47.0 and later.
allowUserCertBasedAuthenticationWithOcspValidation	boolean	If enabled (true), authentication certificates are validated using the Online Certificate Status Protocol (OCSP) or a Certificate Revocation List (CRL).
canConfirmEmailChangeInLightningCommunities	boolean	If **Require email confirmations for email address changes** is enabled (true), when users change their email address, they receive an email at the new address with a link. After they click the link, their new email address takes effect. For orgs created before Winter '20, the field isn't enabled by default. For new orgs, this field is always enabled. To disable the field (not recommended), contact Salesforce Customer Support. Available in API version 47.0 and later.
canConfirmIdentityBySmsOnly	boolean	Prevents identity verification by email for users who have registered other verification methods, such as SMS or Salesforce Authenticator. If no other verification methods are configured, users are verified by email. By default, this setting is disabled (false) for existing orgs. For new orgs, this setting is enabled (true) by default. Available in API version 48.0 and later.
disableTimeoutWarning	boolean	Indicates whether the session timeout warning popup is disabled (true) or enabled (false).
enableBuiltInAuthenticator	boolean	Indicates whether users can verify their identity with a built-in authenticator that's already on their device (true), such as Touch ID or Windows Hello, or not (false). The default value is false.
enableCSPOnEmail	boolean	Indicates whether a content security policy is enabled for the email template. A content security policy helps prevent cross-site scripting attacks by listing allowed sources of images and other content.
enableCSRFOnGet	boolean	Indicates whether Cross-Site Request Forgery (CSRF) protection on GET requests on non-setup pages is enabled (true) or disabled (false).
enableCSRFOnPost	boolean	Indicates whether Cross-Site Request Forgery (CSRF) protection on POST requests on non-setup pages is enabled (true) or disabled (false).
enableCacheAndAutocomplete	boolean	Indicates whether the user's browser is allowed to store usernames and auto-fill the User Name field on the login page (true) or not (false).

Field	Field Type	Description
`enableClickjackNonsetupSFDC`	boolean	Indicates whether clickjack protection for non-setup Salesforce pages is enabled (`true`) or disabled (`false`).
`enableClickjackNonsetupUser`	boolean	Indicates whether clickjack protection for customer Visualforce pages with standard headers turned on is enabled (`true`) or disabled (`false`).
`enableClickjackNonsetupUserHeaderless`	boolean	Indicates whether clickjack protection for customer Visualforce pages with standard headers turned off is enabled (`true`) or disabled (`false`). Available in API version 34.0 and later.
`enableClickjackSetup`	boolean	Indicates whether clickjack protection for setup pages is enabled (`true`) or disabled (`false`).
`enableContentSniffingProtection`	boolean	Indicates whether the browser is prevented from inferring the MIME type from the document content and from executing malicious files (JavaScript, Style sheet) as dynamic content. This field is available in API version 39.0 and later. In API version 58.0 and later, `enableContentSniffingProtection` is always `true`.
`enableLightningLogin`	boolean	If enabled (`true`), users can use Lightning Login (Salesforce Authenticator) to log in instead of a password. Available in API Version 47.0 and later.
`enableLightningLoginOnlyWithUserPerm`	boolean	If enabled (`true`), only users with the Lightning Login User permission can log in with Salesforce Authenticator instead of a password. Available in API version 47.0 and later.
`enableMFADirectUILoginOptIn`	boolean	Requires all users in your Salesforce org to provide an additional verification method when logging in directly to the UI with their username and password. Users who are already enabled via the Multi-Factor Authentication for User Interface Logins user permission experience no change. The Waive Multi-Factor Authentication for Exempt Users user permission overrides this setting.
`enableOauthCorsPolicy`	boolean	If set to `true`, enables Cross-Origin Resource Sharing (CORS) for these OAuth endpoints: • `/services/oauth2/token` • `/services/oauth2/revoke` • `/services/oauth2/introspect` Default setting is `false`. Available in API version 50.0 and later.
`enablePostForSessions`	boolean	Indicates whether cross-domain session information is exchanged using a POST request instead of a GET request, such as when a user is using a Visualforce page. In this context, POST

Field	Field Type	Description
		requests are more secure than GET requests. Available in API version 31.0 and later.
`enableSMSIdentity`	boolean	If enabled (`true`), the default, users can receive a one-time password in a text message (SMS) to verify their identity. Users must verify their mobile phone number before they can receive SMS messages.
`enableU2F`	boolean	If enabled (`true`), users can use a physical U2F-compatible security key for multi-factor authentication (MFA) and identity verification. The default is `false`. Available in API version 47.0 and later.
`enableUpgradeInsecureRequests`	boolean	Indicates whether HTTPS is required for connecting to third-party domains. This setting is enabled by default on accounts created after the Summer '17 release. This field is available in API version 42.0 to 60.0.
`enableXssProtection`	boolean	Indicates whether the HTTP `X-XSS-Protection` response header is enabled to protect against reflected cross-site scripting attacks. This field is available in API version 39.0 to 59.0. The HTTP `X-XSS-Protection` response header is deprecated. To help prevent cross-site scripting (XSS) and other code injection attacks, use the CSPTrustedSite metadata type.
`enforceIpRangesEveryRequest`	boolean	If `true`, the IP addresses in Login IP Ranges are enforced when a user accesses Salesforce (on every page request), including access from a client app. If `false`, the IP addresses in Login IP Ranges are enforced only when a user logs in. This field affects all user profiles with login IP restrictions. Available in API version 34.0 and later.
`enforceUserDeviceRevoked`	boolean	If enabled, and a UserDevice's status is set to revoked, that device can't log in from a Salesforce app. Logins from browsers aren't affected. This field is available in API version 50.0 and later.
`forceLogoutOnSessionTimeout`	boolean	If enabled (`true`), the default, when sessions time out for inactive users, current sessions become invalid. The browser refreshes and returns to the login page. To access the org, the user must log in again. Available in API version 31.0 and later.
`forceRelogin`	boolean	If `true`, an admin who is logged in as another user must log in again to their original session, after logging out as the

Field	Field Type	Description
		secondary user. If `false`, the admin isn't required to log in again.
`hasRetainedLoginHints`	boolean	If you enable **Remember me until logout** (`true`), usernames (login hints) are cached until the user logs out. If a session times out, usernames appear on the Switcher as inactive. If `false` (default), usernames aren't cached for SSO sessions.
`hasUserSwitching`	boolean	If **Enable user switching** is`true` (default), users can log in to other orgs by selecting their profile picture and using the Switcher. You must also enable the **Enable caching and autocomplete on login page** setting. If `false`, the Switcher isn't enabled and your org doesn't appear in Switchers on other orgs.
`hstsOnForcecomSites`	boolean	Indicates whether Visualforce, Salesforce sites, or Experience Cloud sites must use HTTPS. Available in API version 41.0 and later.
`identityConfirmationOnEmailChange`	boolean	Indicates whether a user's identity is confirmed when changing their email address, instead of requiring a relogin. This field is available in API version 42.0 and later.
`identityConfirmationOnTwoFactorRegistrationEnabled`	boolean	Indicates whether users are required to confirm their identities when adding a verification method such as Salesforce Authenticator for multi-factor authentication (MFA), instead of requiring a relogin. (Multi-factor authentication was formerly called two-factor authentication.) This field is available in API version 40.0 and later.
`lockSessionsToDomain`	boolean	Indicates whether the current UI session for a user is associated with a specific domain. This check helps prevent unauthorized use of the session ID in another domain. The value is `true` by default for orgs created with the Spring '15 release or later. Available in API version 33.0 and later.
`lockSessionsToIp`	boolean	Indicates whether user sessions are locked to the IP address from which the user logged in (`true`) or not (`false`).
`lockerServiceAPIVersion`	string	The API version that Lightning Locker enforces for security of custom Lightning components. The default value matches the Salesforce API version of the current release. Only valid Salesforce API versions between 46.0 and the current release can be specified. The version must be specified as a string in the format `"nn.0"`, such as `"48.0"`. This setting has no effect on the `lockerServiceNext` setting, which enables Lightning Web Security. This field is available in API version 47.0 and later.

Field	Field Type	Description
`lockerServiceCSP`	boolean	If `true`, a stricter Content Security Policy is enabled to disallow the `unsafe-inline` source for the `script-src` CSP directive. Script tags can't be used to load JavaScript, and event handlers can't use inline JavaScript. Lightning Locker and Lightning Web Security depend on this setting to be enabled to protect Lightning components.
`lockerServiceNext`	boolean	If `true`, Lightning Web Security is used instead of Lightning Locker to protect Lightning web components. Lightning Locker continues to protect Aura components. If `false`, Lightning Locker protects Lightning web components and Aura components. Available in API version 53.0 and later.
`lockerServiceNextControl`	boolean	Reserved for internal use.
`logoutURL`	string	The URL to which users are redirected when they log out of Salesforce. If no value is specified, the default is `https://`*`MyDomainName`*`.my.salesforce.com`. Available in API version 34.0 and later.
`redirectBlockModeEnabled`	boolean	If `true`, users can't access untrusted URLs outside the Salesforce domains via links in URL or Long Text Area fields. When a user who accesses Salesforce via Salesforce Classic clicks the link, a message informs the user that they can't access the page because the external site isn't trusted. The default is `false`. To specify the URLs that you trust, use the RedirectWhitelistUrl Metadata type. The `redirectBlockModeEnabled` and `redirectionWarning` fields are mutually exclusive. Only one of those fields can be `true`. Available in API 56.0 and later.
`redirectionWarning`	boolean	If `true`, users who accesses Salesforce via Salesforce Classic see an alert when they click a link in a URL or Long Text Area field that redirects them to an untrusted URL outside the Salesforce domains. The default is `true` in orgs created in Spring '18 and later and `false` in orgs created in Winter '18 and earlier. To specify the URLs that you trust, use the RedirectWhitelistUrl Metadata type. The `redirectBlockModeEnabled` and `redirectionWarning` fields are mutually exclusive. Only one of those fields can be `true`. Available in API version 42.0 and later.

Field	Field Type	Description
`referrerPolicy`	boolean	If `true`, pages served by Salesforce for this org include the `referrer-policy` HTTP header with the directive defined by `referrerPolicyDirective`. If `false`, that HTTP header isn't included and requests can always see the full URL of the Salesforce page. The default is `true`. Available in API version 42.0 and later. In API version 42.0–57.0, if `referrerPolicy` is `true`, pages served by Salesforce for this org include the `referrer-policy` HTTP header with the `origin-when-cross-origin` directive.
`referrerPolicyDirective`	ReferrerPolicy (enumeration of type string)	The HTTP referrer policy directive for pages served by Salesforce. The default is `origin-when-cross-origin`. If `referrerPolicy` is `false`, this value has no effect. Available in API version 58.0 and later. Valid values are: • `origin-when-cross-origin`—Send the origin only for cross-domain requests or when the target website is on a downgraded protocol. An example of a downgraded protocol is a request made from an HTTPS URL to an HTTP site. This is the default setting. • `no-referrer`—Never include the referrer. • `no-referrer-when-downgrade`—Omit the referrer when the target website is on a downgraded protocol. For example, when making a request to an HTTP site from an HTTPS URL. This `referrerPolicyDirective` isn't recommended,because the full URL of the page is exposed to cross-origin requests to the same or a higher protocol level. For example, requests from HTTPS to HTTPS and requests from HTTP to either HTTP or HTTPS. • `origin`—Always send the origin only. • `same-origin`—Omit the referrer for cross-origin requests. • `strict-origin`—For requests with the same protocol level (HTTPS to HTTPS), send the origin only. Omit the referrer when the target website is on a downgraded protocol. An example of a downgraded protocol is a request made from an HTTPS URL to an HTTP site. • `strict-origin-when-cross-origin`—For same-origin requests, send the full referrer URL. For cross-origin requests with the same protocol level (HTTPS to HTTPS), send the origin only. Omit the referrer when the target website is on a downgraded protocol.

Field	Field Type	Description
		• `unsafe-url`—Always include the full referrer URL. This referrerPolicyDirective isn't recommended because the full URL of the page is exposed to requests from insecure origins. For more information on HTTP referrer policy directives, including examples, see the Referrer-Policy entry in the *MDN Docs HTTP Guide*.
`requireHttpOnly`	boolean	Sets the `HttpOnly` attribute on session cookies, making them inaccessible via JavaScript. If `true`, session ID cookie access is restricted. If `false`, access is restricted. If you have a custom or packaged application that uses JavaScript to access session ID cookies, your application breaks if `requireHttpOnly` is set to `true`. The application can't access the cookie. This field is available in API version 40.0 to 60.0.
`requireHttps`	boolean	Determines whether HTTPS is required to log in to or access Salesforce. This option is enabled by default for security reasons and can't be disabled. To change to HTTP, contact Salesforce Customer Support. This field is available in API version 40.0 and later.
`securityCentralKillSession`	boolean	Deprecated in API version 36.0 to 50.0. Removed in API version 51.0 and later.
`sessionTimeout`	SessionTimeout (enumeration of type string)	The length of time after which users without activity are prompted to log out or continue working. Valid values are: • `FifteenMinutes` • `ThirtyMinutes` • `SixtyMinutes` • `NinetyMinutes`—Available in API version 58.0 and later. • `TwoHours` • `FourHours` • `EightHours` • `TwelveHours` • `TwentyFourHours`—Available in API version 38.0 and later.

Field	Field Type	Description
`sidToken3rdPartyAuraApp`	boolean	If `true`, a Lightning app replaces the authentication cookie with a session token when the Lightning app is in a third-party context, such as Lightning Out. Browsers are restricting the use of third-party cookies. This org setting is an alternative for the authentication cookie to requiring that users disable browser settings, such as Safari's `Prevent cross-site tracking` setting. This field is available in API version 59.0 and later.
`skipSFAWhenMFADirectUILogin`	boolean	Indicates which screen users see first when they're prompted to register a verification method for multi-factor authentication (MFA). If `true`, users see a list of all supported verification methods. If `false`, users see only the Salesforce Authenticator option. To see a list of all supported verification methods, users must navigate to a new page.
`terminateUserSessionsWhenAdminResetsPassword`	boolean	Indicates what happens to a user's UI sessions when an admin resets that user's password. If `true`, all of the user's UI sessions are terminated. If `false`, no UI sessions are terminated.
`untrustedRedirect`	untrustedRedirect (enumeration of type string)	The redirection behavior when a user who accesses Salesforce via Lightning Experience clicks a hyperlink in a URL field with a target URL that isn't trusted. Valid values are: • `AlwaysAllowed` • `NeverAllowed` • `WithUserPermission` To specify the URLs that you trust, use the RedirectWhitelistUrl Metadata type. Available in API version 64.0 and later. Secure redirections to untrusted URLs in Lightning Experience is a pilot or beta service that is subject to the Beta Services Terms at Agreements - Salesforce.com or a written Unified Pilot Agreement if executed by Customer, and applicable terms in the Product Terms Directory. Use of this pilot or beta service is at the Customer's sole discretion.
`useLocalStorageForLogoutUrl`	boolean	Redirects all expired tabs in your browser to your custom logout URL (`true`). By default, this option is enabled for all new orgs and is available in API version 52.0 and later. For orgs created prior to the Summer '21 release, the default setting is `false`. Before enabling this setting, review these considerations.

Field	Field Type	Description
		• This setting uses the browser's local storage to store the custom logout URL. • Verify that this setting doesn't interfere with your custom login integrations.
`welcomeEmailTemplateId`	string	Custom email template for the welcome email that new internal users receive when they're registered. This field supports only Classic email templates. Available in API version 63.0 and later.

SingleSignOnSettings

Represents your org's single sign-on (SSO) settings. These settings are available API version 47.0 and later.

Field Name	Field Type	Description
`enableCaseInsensitiveFederationID`	boolean	If you enable **Make Federation ID case-insensitive** (`true`), the Federation ID field on a user object isn't case-sensitive. If disabled (`false`), the Federation ID field remains case-sensitive. The default is `false`.
`enableForceDelegatedCallout`	boolean	If you enable **Force Delegated Authentication Callout** (`true`), a callout to the SSO endpoint occurs regardless of login restriction failures. If disabled (`false`), the default, and if a user's first login attempt fails due to login restrictions within the Salesforce org, a call isn't made to the SSO endpoint.
`enableMultipleSamlConfigs`	boolean	If `true` (default), you can configure multiple SAML providers. After enabling the setting, it can't be disabled.
`enableSamlJitProvisn'tioning`	boolean	If you enable **User Provisioning Enabled** (`true`), you can provision users through a SAML assertion (called just-in-time provisioning). Requires `EnableSamlLogin` to be `true` and `enableMultipleSamlConfigs` to be `false`. The default is enabled (`false`).
`enableSamlLogin`	boolean	If you enable **SAML Enabled** (`true`), users can SSO into Salesforce from providers via SAML. The default isn't enabled (`false`).
`isLoginWithSalesforceCredentialsDisabled`	boolean	If **Disable login with Salesforce credentials** is `true`, users are redirected to third-party identity providers for authentication. The default is enabled (`false`). If you enabled this feature prior to the Summer '20 release and want to disable it prior to July 27, 2020, contact Customer Support.

Declarative Metadata Sample Definition

The following is a sample `security.settings` metadata file.

```
<?xml version="1.0" encoding="UTF-8"?>
<SecuritySettings xmlns="http://soap.sforce.com/2006/04/metadata"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <canUsersGrantLoginAccess>true</canUsersGrantLoginAccess>
    <enableAdminLoginAsAnyUser xsi:nil="true"/>
    <enableAuditFieldsInactiveOwner xsi:nil="true"/>
    <enableAuraSecureEvalPref xsi:nil="true"/>
    <enableCoopHeader>true</enableCoopHeader>
    <enableCoepHeader>false</enableCoepHeader>
    <enableCrossOrgRedirects>false</enableCrossOrgRedirects>
    <enablePermissionsPolicy>true</enablePermissionsPolicy>
    <grantCameraAccess>TrustedUrls</grantCameraAccess>
    <grantMicrophoneAccess>TrustedUrls</grantMicrophoneAccess>
    <networkAccess/>
    <passwordPolicies>
        <complexity>NoRestriction</complexity>
        <expiration>Never</expiration>
        <historyRestriction>0</historyRestriction>
        <lockoutInterval>FifteenMinutes</lockoutInterval>
        <maxLoginAttempts>TenAttempts</maxLoginAttempts>
        <minimumPasswordLength>5</minimumPasswordLength>
        <minimumPasswordLifetime>false</minimumPasswordLifetime>
        <obscureSecretAnswer>false</obscureSecretAnswer>
        <questionRestriction>DoesNotContainPassword</questionRestriction>
    </passwordPolicies>
    <redirectBlockModeEnabled>false</redirectBlockModeEnabled>
    <sendCspForUncommonClients>false</sendCspForUncommonClients>
    <sessionSettings>
       <allowUserAuthenticationByCertificate>false</allowUserAuthenticationByCertificate>

        <disableTimeoutWarning>false</disableTimeoutWarning>
        <enableBuiltInAuthenticator>false</enableBuiltInAuthenticator>
        <enableCSPOnEmail>true</enableCSPOnEmail>
        <enableCSRFOnGet>true</enableCSRFOnGet>
        <enableCSRFOnPost>true</enableCSRFOnPost>
        <enableCacheAndAutocomplete>true</enableCacheAndAutocomplete>
        <enableClickjackNonsetupSFDC>true</enableClickjackNonsetupSFDC>
        <enableClickjackNonsetupUser>false</enableClickjackNonsetupUser>
       <enableClickjackNonsetupUserHeaderless>false</enableClickjackNonsetupUserHeaderless>

        <enableClickjackSetup>true</enableClickjackSetup>
        <enableContentSniffingProtection>true</enableContentSniffingProtection>
        <enableLightningLogin>true</enableLightningLogin>
       <enableLightningLoginOnlyWithUserPerm>false</enableLightningLoginOnlyWithUserPerm>

        <useLocalStorageForLogoutUrl>false</useLocalStorageForLogoutUrl>
        <enableOauthCorsPolicy>false</enableOauthCorsPolicy>
        <enablePostForSessions>false</enablePostForSessions>
        <enableSMSIdentity>true</enableSMSIdentity>
        <enableU2F>false</enableU2F>
        <enforceIpRangesEveryRequest>false</enforceIpRangesEveryRequest>
```

```
        <enforceUserDeviceRevoked>false</enforceUserDeviceRevoked>
        <forceLogoutOnSessionTimeout>true</forceLogoutOnSessionTimeout>
        <forceRelogin>true</forceRelogin>
        <hasRetainedLoginHints>false</hasRetainedLoginHints>
        <hasUserSwitching>true</hasUserSwitching>
        <hstsOnForcecomSites>false</hstsOnForcecomSites>
        <identityConfirmationOnEmailChange>true</identityConfirmationOnEmailChange>

<identityConfirmationOnTwoFactorRegistrationEnabled>true</identityConfirmationOnTwoFactorRegistrationEnabled>

        <lockSessionsToDomain>true</lockSessionsToDomain>
        <lockSessionsToIp>false</lockSessionsToIp>
        <lockerServiceAPIVersion>56.0</lockerServiceAPIVersion>
        <lockerServiceCSP>true</lockerServiceCSP>
        <lockerServiceNext>true</lockerServiceNext>
        <logoutURL>https://mycompany.com</logoutUrl>
        <redirectionWarning>true</redirectionWarning>
        <referrerPolicy>true</referrerPolicy>
       <referrerPolicyDirective>strict-origin-when-cross-origin</referrerPolicyDirective>
        <requireHttps>false</requireHttps>
        <sessionTimeout>TwoHours</sessionTimeout>
        <untrustedRedirect>WithUserPermission</untrustedRedirect>
        <useLocalStorageForLogoutUrl>true</useLocalStorageForLogoutUrl>
        <welcomeEmailTemplateId>X000000000000</welcomeEmailTemplateId>
    </sessionSettings>
    <singleSignOnSettings>
        <enableCaseInsensitiveFederationID>false</enableCaseInsensitiveFederationID>
        <enableForceDelegatedCallout>false</enableForceDelegatedCallout>
        <enableMultipleSamlConfigs>true</enableMultipleSamlConfigs>
        <enableSamlJitProvisioning>false</enableSamlJitProvisioning>
        <enableSamlLogin>false</enableSamlLogin>

<isLoginWithSalesforceCredentialsDisabled>true</isLoginWithSalesforceCredentialsDisabled>

    </singleSignOnSettings>
</SecuritySettings>
```

The following is an example `package.xml` manifest that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>Security</members>
        <name>Settings</name>
    </types>
    <version>64.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ServiceCloudVoiceSettings

Represents an organization's Service Cloud Voice settings. This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

ServiceCloudVoiceSettings values are stored in the `ServiceCloudVoice.settings` file in the `settings` folder. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

ServiceCloudVoiceSettings is available in API versions 52.0 and later.

Fields

Field Name	Field Type	Description
`enableSCVExternalTelephony`	boolean	Indicates whether to allow a third-party telephony service to work with Service Cloud Voice with Partner Telephony.
`enableServiceCloudVoice`	boolean	Indicates whether to enable the Service Cloud Voice with Amazon Connect.
`enableVoiceInGovCloudOptIn`	boolean	Indicates whether you agree to the terms of using Service Cloud Voice with Amazon Connect in a Salesforce Government Cloud environment. Amazon Connect is a third-party Amazon service that sits outside the Salesforce Government Cloud FedRAMP environment. Amazon Connect is a separate service offered by Amazon and not a FedRAMP authorized service. Therefore, Amazon Connect's processing environment falls outside the Government Cloud FedRAMP authorization boundary. To learn more, see Amazon Connect.

Declarative Metadata Sample Definition

The following is an example of a ServiceCloudVoice.settings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceCloudVoiceSettings xmlns="http://soap.sforce.com/2006/04/metadata">
 <enableServiceCloudVoice>true</enableServiceCloudVoice>
</ServiceCloudVoiceSettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>ServiceCloudVoice</members>
```

```
        <name>Settings</name>
    </types>
    <version>52.0</version>
</Package>
```

ServiceSetupAssistantSettings

Represents an organization's Service Setup Assistant settings. The Service Setup Assistant can be used to set up a basic service console app.

This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

ServiceSetupAssistantSettings values are stored in the `ServiceSetupAssistant.settings` file in the `settings` directory. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

ServiceSetupAssistantSettings components are available in API version 50.0 and later.

Fields

Field Name	Field Type	Description
`enableServiceSetupAssistant`	boolean	Indicates whether the Service Setup Assistant is enabled (`true`) or not (`false`).

Declarative Metadata Sample Definition

The following is an example of a ServiceSetupAssistantSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceSetupAssistantSettings xmlns="http://soap.sforce.com/2006/04/metadata">
 <enableServiceSetupAssistant>true</enableServiceSetupAssistant>
</ServiceSetupAssistantSettings>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SharingSettings

Represents an organization's sharing, visibility, and data access settings. This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

SharingSettings values are stored in the `Sharing.settings` file in the `settings` directory. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

SharingSettings is available in API version 47.0 and later.

Special Access Rules

To use SharingSettings, you need the Manage Sharing permission.

Fields

Field Name	Field Type	Description
`deferGroupMembership`	boolean	Indicates whether group membership calculations are suspended (`true`) or not (`false`). This field has a default value of `false`. This field is available in API version 49.0 and later. **Important:** • The defer sharing calculation feature isn't enabled by default. To enable it for your Salesforce org, contact Salesforce Customer Support. • When you change the value of this field from `true` to `false`, group membership is automatically recalculated. Sharing rules are also automatically recalculated, unless the `deferSharingRules` field is set to `true` prior to modifying `deferGroupMembership`. Depending on your org, these recalculations can take a significant amount of time to complete. • If the `deferGroupMembership` field is set to `true`, you can't change the value of `deferSharingRules`. Sharing rule calculations are suspended regardless of the value of `deferSharingRules`.
`deferSharingRules`	boolean	Indicates whether sharing rule calculations are suspended (`true`) or not (`false`). This field has a default value of `false`. This field is available in API version 49.0 and later. **Important:** • The defer sharing calculation feature isn't enabled by default. To enable it for your Salesforce org, contact Salesforce Customer Support.

Field Name	Field Type	Description
		• When you change the value of this field from `true` to `false`, sharing rules are automatically recalculated. Depending on your org, this recalculation can take a significant amount of time to complete. • If the `deferGroupMembership` field is set to `true`, you can't change the value of `deferSharingRules`. Sharing rule calculations are suspended regardless of the value of `deferSharingRules`.
`enableAccountRoleOptimization`	boolean	Indicates whether person roles are assigned to new site users in accounts without existing users (`true`) or if regular site roles are created for new users (`false`). This field has a default value of `false`.
`enableAssetSharing`	boolean	Indicates whether sharing is enabled for assets (`true`) or asset access is determined by the parent object's sharing rules (`false`). This field has a default value of `false`.
`enableCommunityUserVisibility`	boolean	Indicates whether site users in the same site can see each other regardless of the organization-wide defaults (`true`) or not (`false`). This field has a default value of `false`. In orgs created in API version 47.0 and later, this setting doesn't apply to guest users.
`enableExternalSharingModel`	boolean	Indicates whether the external sharing model is enabled (`true`) or not (`false`). This field has a default value of `true` if Salesforce Experiences are enabled, and a default value of `false` if not. To use this field, you need the Customize Application permission.
`enableManagerGroups`	boolean	Indicates whether users can share records with their managers and manager subordinates groups (`true`) or not (`false`). This field has a default value of `false`. To use this field, you need the View and Manage Users permission.
`enableManualUserRecordSharing`	boolean	Indicates whether users can share their own user record (`true`) or not (`false`). This field has a default value of `false`.
`enablePartnerSuperUserAccess`	boolean	Indicates whether you can grant super user access to partners in sites (`true`) or not (`false`). This field has a default value of `false`. To use this field, you need the Customize Application permission
`enablePortalUserCaseSharing`	boolean	Indicates whether portal users can access related contacts for cases that they own (`true`) or not (`false`). This field has a default value of `false`.
`enablePortalUserVisibility`	boolean	Indicates whether portal users in the same customer or partner portal account can see each other regardless of the organization-wide defaults (`true`) or not (`false`). This field has a default value of `false`. To enable this field, contact Salesforce Support.
`enableRemoveTMGroupMembership`	boolean	Removes group membership info for the original territory management feature after migrating to Sales Territories when set to `true`. This field

Field Name	Field Type	Description
		has a default value of `false`. Once this field is set to `true`, it can't be set to `false` again.
`enableRestrictAccessLookupRecords`	boolean	Indicates whether users must have read access to a record to see the record's name in lookup and system fields (`true`) or not (`false`). This field has a default value of `true` in Salesforce orgs created in Spring '20 or later and a default value of `false` in all other orgs. This field is available in API version 48.0 and later.
`enableSecureGuestAccess`	boolean	When `true`, guest users have org-wide defaults set to Private. To share records with them, you must use guest user sharing rules. As of API version 50.0, this field's value is always `true`, regardless of the value that you set. Changing its value has no effect on Salesforce, even if it reads `false`. This change applies retroactively back to API version 47.0, when this field was first introduced. Previously, in API version 47.0 to 49.0, this field indicated whether guest users' record access is secured (`true`) or not (`false`), and the field's default value was `false`. Now, in all API versions, this field's value is always `true`, even if it reads `false`.
`enableStandardReportVisibility`	boolean	Indicates whether users can view reports based on standard report types that may expose data of users to whom they don't have access (`true`) or not (`false`). This field has a default value of `false`.
`enableTerritoryForecastManager`	boolean	Indicates whether forecast managers can act as delegated administrators for territories below them in the hierarchy (`true`) or not (`false`). This field has a default value of `false`.

Declarative Metadata Sample Definition

The following is an example of a SharingSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<SharingSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <deferGroupMembership>false</deferGroupMembership>
    <deferSharingRules>false</deferSharingRules>
    <enableAccountRoleOptimization>false</enableAccountRoleOptimization>
    <enableAssetSharing>false</enableAssetSharing>
    <enableCommunityUserVisibility>false</enableCommunityUserVisibility>
    <enableExternalSharingModel>true</enableExternalSharingModel>
    <enableManagerGroups>false</enableManagerGroups>
    <enableManualUserRecordSharing>true</enableManualUserRecordSharing>
    <enablePartnerSuperUserAccess>false</enablePartnerSuperUserAccess>
    <enablePortalUserCaseSharing>false</enablePortalUserCaseSharing>
    <enablePortalUserVisibility>true</enablePortalUserVisibility>
    <enableRemoveTMGroupMembership>false</enableRemoveTMGroupMembership>
    <enableRestrictAccessLookupRecords>true</enableRestrictAccessLookupRecords>
    <enableSecureGuestAccess>true</enableSecureGuestAccess>
    <enableStandardReportVisibility>false</enableStandardReportVisibility>
```

```
    <enableTerritoryForecastManager>false</enableTerritoryForecastManager>
</SharingSettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>Sharing</members>
        <name>Settings</name>
    </types>
    <version>47.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SiteSettings

Represents the settings for Experience Cloud sites and for Salesforce Sites.

This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

SiteSettings values are stored in a single file named `Site.settings` in the `settings` directory. The `.settings` files are different from other named components because there's only one `.settings` file for each settings component.

Version

SiteSettings components are available in API version 47.0 and later.

Fields

Field Name	Field Type	Description
enableEnhancedSitesAndContentPlatform	boolean	Indicates whether the enhanced sites and content platform for Experience Cloud is enabled for your org (`true`) or not (`false`). The default is `true`. When `true`, new LWR sites and enhanced CMS workspaces are hosted together on a redesigned platform that offers partial deployment, site content search, and easy content management. Enhanced LWR sites are represented by the DigitalExperienceBundle and DigitalExperienceConfig types. Available in API version 56.0 and later.

Field Name	Field Type	Description
enableExpBuilderCopilot	boolean	Enables Agentforce (beta) in Experience Builder for enhanced LWR sites. The default value is false. Available in API 64.0 and later. Note: Agentforce Experience Builder Agent is a pilot or beta service that is subject to the Beta Services Terms at Agreements - Salesforce.com or your written Unified Pilot Agreement, and following terms in the Product Terms Directory: Non-GA Agentforce, the Non-GA Open AI LLM Provider and the Non-GA Credit Consumption. Use of this pilot or beta service consumes Einstein Requests and is at the Customer's sole discretion.
enableExperienceFriendlyUrls	boolean	Indicates whether SEO-friendly URL snippets, or "slugs," are enabled for your org (true) or not (false). The default is false. When true, available only in B2C Commerce LWR sites. Available only in API version 58.0. In API version 59.0 and later, use expFriendlyUrlsAsDefault in the Network type.
enableProxyLoginICHeader	boolean	Indicates whether security tokens for API logins from callouts (in API version 31.0 and earlier) are required (true) or not (false). The default value is true.
enableSitesRecordReassignOrgPref	boolean	Deprecated in API version 63.0 and later. When true, indicates when the org assigns records created by guest users of a site to a default owner in the org. When false, the guest user remains the owner of the record. The default value is false. Available in API version 48.0 through 63.0.
enableTopicsInSites	boolean	Indicates whether guest and authenticated external users can view topics in Salesforce Sites and Salesforce portals (true) or not (false). The default value is false.
enableVisualforceApiAccessAllowed	boolean	Deprecated in API version 52.0 and later. Allow users of Visualforce pages to override API access control restrictions and access APIs when the enableAdminApprovedAppsOnly in ConnectedAppSettings is enabled (true). The default value is false.
enableWebruntimeBYOTemplate	boolean	Indicates whether the Build Your Own (LWC) template is available in Experience Builder. The default value is false. Available in API version 48.0 and later. Removed in API version 51.0.

Declarative Metadata Sample Definition

The following is an example of a SiteSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<SiteSettings xmlns="http://soap.sforce.com/2006/04/metadata">
     <enableProxyLoginICHeader>true</enableProxyLoginICHeader>
     <enableTopicsInSites>false</enableTopicsInSites>
</SiteSettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>Site</members>
        <name>Settings</name>
    </types>
    <version>47.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SocialCustomerServiceSettings

Represents Social Customer Service settings such as how to format inbound content from social posts to cases. This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

SocialCustomerServiceSettings components have the suffix `settings` and are stored in the `settings` folder. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

SocialCustomerServiceSettings is available in API version 41.0 and later.

Fields

Field Name	Field Type	Description
`caseSubjectOption`	CaseSubjectOption (enumeration of type string)	Required. Specifies an option from which inbound social content is formatted to appear in case records' **Case Subject** field. Valid values are: • `SocialPostSource` • `SocialPostContent` • `BuildCustom`
`enableAllFBResponseAccounts`	boolean	Indicates whether responses from all Facebook managed accounts are enabled. If this setting is disabled, responses to a Facebook post can only be sent from the account that the original customer post was directed to. The default value is `true`. Available in API version 56.0 and later.

Field Name	Field Type	Description
`enableSocialApprovals`	boolean	Indicates whether social approvals are enabled. To learn more, see Enable Social Post Approvals.The default value is `false`. Available in API version 47.0 and later.
`enableSocialCaseAssignmentRules`	boolean	Indicates whether case assignment rules are enabled. Use case assignment rules to determine how cases are assigned to users or put into queues as they are created. The default value is `false`. Available in API version 47.0 and later.
`enableSocialCustomerService`	boolean	Indicates whether to enable the Social Customer Service feature. The default value is `false`. Available in API version 47.0 and later.
`enableSocialPersonaHistoryTracking`	boolean	Indicates whether to enable Social Persona history tracking. History tracking helps identify who made what changes when, and for differentiating between automatic and manual changes. The default value is `false`. Available in API version 47.0 and later.
`enableSocialPostHistoryTracking`	boolean	Indicates whether to enable Social Post history tracking. History tracking helps identify who made what changes when, and for differentiating between automatic and manual changes. The default value is `false`. Available in API version 47.0 and later.
`enableSocialReceiveParentPost`	boolean	Indicates whether to use the original social post that initiated the case as the parent record. The default value is `false`. Available in API version 47.0 and later.

Declarative Metadata Sample Definition

This is a sample of a `SocialCustomerServiceSettings.settings` file.

```
<?xml version="1.0" encoding="UTF-8"?>
<SocialCustomerServiceSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <caseSubjectOption>SocialPostSource</caseSubjectOption>
    <enableSocialApprovals>true</enableSocialApprovals>
    <enableSocialCaseAssignmentRules>false</enableSocialCaseAssignmentRules>
    <enableSocialCustomerService>true</enableSocialCustomerService>
    <enableSocialPersonaHistoryTracking>false</enableSocialPersonaHistoryTracking>
    <enableSocialPostHistoryTracking>false</enableSocialPostHistoryTracking>
    <enableSocialReceiveParentPost>true</enableSocialReceiveParentPost>
</SocialCustomerServiceSettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>SocialCustomerService</members>
        <name>Settings</name>
```

```
    </types>
    <version>47.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SocialProfileSettings

Represents org preferences for social media features such as enabling Twitter and Facebook.Represents org preferences for social media features such as enabling Twitter and Facebook. This type extends the Metadata metadata type and inherits the fullName field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

SocialProfileSettings values are stored in a single file named `SocialProfile.settings` in the `settings` directory of the corresponding package directory. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

SocialProfileSettings is available in API version 47.0 and later.

Fields

Field Name	Field Type	Description
`isFacebookSocialProfilesDisabled`	boolean	Prevents users from accessing Facebook in social CRM (`true`) or not (`false`). `enableSocialProfiles` must be `true` to enable Facebook social profiles.
`isLinkedInSocialProfilesDisabled`	boolean	Prevents users from accessing LinkedIn in social CRM (`true`) or not (`false`). `enableSocialProfiles` must be `true` to enable LinkedIn social profiles.
`isTwitterSocialProfilesDisabled`	boolean	Prevents users from accessing Twitter in social CRM (`true`) or not (`false`). `enableSocialProfiles` must be `true` to enable Twitter social profiles. This setting is permanently set to True because Twitter access was removed in API version 59.0.

Field Name	Field Type	Description
`isYouTubeSocialProfilesDisabled`	boolean	Prevents users from accessing YouTube in social CRM (`true`) or not (`false`). `enableSocialProfiles` must be `true` to enable YouTube social profiles. This setting is permanently set to True because YouTube access was removed in API version 60.0.
`enableSocialProfiles`	boolean	Indicates whether users can access social media profiles in social CRM (`true`) or not (`false`).

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SourceTrackingSettings (Beta)

Represents settings for source tracking, so that changes you make in your Developer and Developer Pro sandboxes or local workspace can be tracked. This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

SourceTrackingSettings values are stored in the `SourceTracking.settings` file in the `settings` folder. The `.settings` files are different from other named components because there is only one settings file for each settings component.

Version

SourceTrackingSettings is available in API version 49.0 and later.

Fields

Field Name	Field Type	Description
`enableSourceTrackingSandboxes`	boolean	Indicates whether to enable source tracking automatically when Developer or Developer Pro sandboxes are created or refreshed (`true`) or not (`false`). The default value is `false`. If you set `enableSourceTrackingSandboxes` back to `false` after it was enabled, a sandbox that is tracking source changes continues to do so until it is refreshed. Note: You don't need to have the Developer Hub (DevHub) enabled in the same org to enable source tracking. This field applies to production orgs only; in other orgs, this field is ignored.

Declarative Metadata Sample Definition

The following is an example of a SourceTrackingSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<SourceTrackingSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <enableSourceTrackingSandboxes>true</enableSourceTrackingSandboxes>
</SourceTrackingSettings>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SubscriptionManagementSettings

Represents the settings used to manage recurring subscriptions.

Parent Type and Manifest Access

This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all the settings metadata types for the org are accessed using the "Settings" name. See Settings for more details.

File Suffix and Directory Location

`SubscriptionManagementSettings` values are stored in the `subscriptionmanagement.settings` file in the `settings` folder. The `.settings` files are different from other named components, because there's only one settings file for each settings component.

Version

SubscriptionManagementSettings components are available in API version 55.0 and later.

Special Access Rules

This metadata type is available with Subscription Management.

Fields

Field Name	Field Type	Description
`enableBillingDocGen`	boolean	Indicates whether document generation is enabled in the org (`true`) or not (`false`). The default value is `false`.

Field Name	Field Type	Description
enableConvert NegativeInvoiceLines ToCreditMemoAndApply	boolean	Indicates whether to convert negative invoice lines into a credit note (true) or not (false). This credit note holds a positive balance that you can later use to apply against future invoices. The default value is false.
enableInvHeaderLvlSettlement	boolean	Indicates whether payments can be applied on the whole invoice (true) or only on invoice lines (false). The default value is false.
enablePaymentScheduleAutomation	boolean	Indicates whether the payment schedule and payment schedule item are created automatically (true) or not (false). The default value is false.
enableRefundAutomation	boolean	Indicates whether refunds are processed automatically (true) or not (false). The default value is false.
enableRevSubMgmtBlngOptOut	boolean	Indicates whether the billing schedules in Subscription Management are disabled (true) or not (false). The default value is false.
enableSubscriptionManagement	boolean	Indicates whether Subscription Management is enabled (true) or not (false). The default value is false.

Declarative Metadata Sample Definition

This example shows a sample SubscriptionManagementSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<SubscriptionManagementSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <enableSubscriptionManagement>true</enableSubscriptionManagement>
</SubscriptionManagementSettings>
```

This example shows a sample package.xml that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
```

```
        <members>SubscriptionManagementSettings</members>
        <name>Settings</name>
    </types>
    <version>64.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SurveySettings

Represents an org's survey settings. Use the SurveySettings component to enable Salesforce Surveys, enable Customer Lifecycle Maps, and choose whether the owner of a survey can manage the responses.

Parent Type and Manifest Access

This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all the settings metadata types for the org are accessed using the "Settings" name. See Settings for more details.

File Suffix and Directory Location

`SurveySettings` values are stored in a single file named `Survey.settings` in the `settings` folder. The `.settings` files are different from other named components because there is only one settings file for each settings component.

Version

SurveySettings is available in API version 47.0 and later.

Fields

Field Name	Field Type	Description
`enableGenerativeAISurveys`	boolean	Indicates whether AI-Generated Surveys is enabled for your org (true) or not (false). The default value is `false`. Available in API version 62.0 and later.
`enableIndustriesCxmEnabled`	boolean	Indicates whether Customer Lifecycle Maps is enabled for your org (true) or not (false). The default value is `false`.
`enableSurvey`	boolean	Indicates whether Surveys is enabled for your org (true) or not (false). The default value is `false`.
`enableSurveyOwnerCanManageResponse`	boolean	Indicates whether the owner of a survey can manage its responses. The default value is `false`.

Declarative Metadata Sample Definition

This example shows a sample SurveySettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<SurveySettingsxmlns="http://soap.sforce.com/2006/04/metadata">
<enableIndustriesCxmEnabled>false</enableIndustriesCxmEnabled>
<enableSurvey>true</enableSurvey>
<enableSurveyOwnerCanManageResponse>false</enableSurveyOwnerCanManageResponse>
<enableGenerativeAISurveys>false</enableGenerativeAISurveys>
</SurveySettings>
```

This example shows a sample `package.xml` file that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>Survey</members>
        <name>Settings</name>
    </types>
    <version>61.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

Territory2Settings

Represents an org's Territory2 settings. Use Territory2 settings to set the access level that Sales Territories users have to records associated with sales territories, and to enable features. The standard record access settings apply to accounts and opportunities. With *`Private`* default internal access for contacts or cases, you can also set access for those records.This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

Territory2Settings values are stored in a single file named `Territory2.settings` in the `settings` directory of the corresponding package directory. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

Territory2Settings is available in API version 32.0 and later.

Special Access Rules

Fields

Field Name	Field Type	Description
`defaultAccountAccessLevel`	string	Sets the default level of access that users have to account records in territories: *`view`* and *`edit`* accounts assigned to territories or *`view`*, *`edit`*, *`transfer`*, and *`delete`* accounts assigned to territories.
`defaultCaseAccessLevel`	string	Sets the default level of access that users have to case records in territories: *`view`* and *`edit`* accounts assigned to territories or *`view`*, *`edit`*, *`transfer`*, and *`delete`* accounts assigned to territories.
`defaultContactAccessLevel`	string	Sets the default level of access that users have to contact records in territories: *`view`* and *`edit`* accounts assigned to territories or *`view`*, *`edit`*, *`transfer`*, and *`delete`* accounts assigned to territories.
`defaultOpportunityAccessLevel`	string	Sets the default level of access that users have to opportunity records in territories: *`view`* and *`edit`* accounts assigned to territories or *`view`*, *`edit`*, *`transfer`*, and *`delete`* accounts assigned to territories.
`enableTerritoryManagement2`	boolean	Enables and disables Sales Territories only. If `true`, Sales Territories is enabled. If `false` (default), Enterprise Territory Management isn't enabled. Enabling and disabling Sales Territories is exclusive of all other operations, and the field value must be `true` before other territory-management operations can run. Available in API version 47.0 and later.
`opportunityFilterSettings`	Territory2SettingsOpportunityFilter	Optional. Specifies an Apex class to assign territories to opportunities and whether you want to run it when an opportunity is created. Available in API version 34.0 and later.
`showTM2EnabledBanner`	boolean	If `true`, a success banner appears on the Territory Settings page in Setup. Available in API version 49.0 and later.
`supportedObjects`	Territory2SupportedObject[]	Sets the user access levels of all objects that support territory assignments in the org. Available in API version 57.0 and later.
`t2ForecastAccessLevel`	string	Sets the access level that users in a parent territory get to the opportunities assigned to its child territories, regardless of who owns the opportunities. Valid values are: • `View` • `Edit` Available in API version 49.0 and later.

Field Name	Field Type	Description
`tm2BypassRealignAccInsert`	boolean	If `true`, account assignment rules don't run during account insert jobs. Available in API version 53.0 and later.
`tm2EnableUserAssignmentLog`	boolean	If `true`, when a user is assigned to a territory, the assignment action is logged. Available in API version 57.0 and later.

Territory2SettingsOpportunityFilter

This subtype specifies an Apex class that assigns territories to opportunities. You can run the Apex class automatically every time a user creates an opportunity, or run it by using multithreading.

Field Name	Description
`apexClassName`	**Field Type** string **Description** Represents the Apex class name.
`enableFilter`	**Field Type** boolean **Description** If `true`, the Apex class is used to assign territories to opportunities.
`runMultiThreaded`	**Field Type** boolean **Description** Required. If `true`, the Apex class runs by using multithreading, which can improve performance. Set this value to `true` only if you're assigning opportunity or opportunity product splits, and your Apex code can run with multithreading. This field has a default value of `false`. Available in API version 62.0 and later.
`runOnCreate`	**Field Type** boolean **Description** If `true`, the Apex class runs automatically every time a user creates an opportunity.

Territory2SupportedObject

Sets the user access levels of all objects that support territory assignments in the org.

Field Name	Description
`defaultAccessLevel`	**Field Type** string **Description** Required. The default user access level as permitted by the organization's sharing settings. Valid values are: • `Read` • `Edit` • `Transfer` • `All`
`objectType`	**Field Type** string **Description** Required. The only supported object type is `Lead`.
`state`	**Field Type** string **Description** Required. Valid values are: • `Disabled` • `Enabled`

Declarative Metadata Sample Definition

The following example shows the definition of a Territory2Settings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<Territory2Settings xmlns="http://soap.sforce.com/2006/04/metadata">
    <defaultAccountAccessLevel>Owner</defaultAccountAccessLevel>
    <defaultOpportunityAccessLevel>Read</defaultOpportunityAccessLevel>
    <defaultCaseAccessLevel>None</defaultCaseAccessLevel>
    <defaultContactAccessLevel>Edit</defaultContactAccessLevel>
    <enableTerritoryManagement2>true</enableTerritoryManagement2>
    <showTM2EnabledBanner>true</showTM2EnabledBanner>
    <supportedObjects>
       <defaultAccessLevel>Read</defaultAccessLevel>
       <state>Disabled</state>
       <objectType>Lead</objectType>
    </supportedObjects>
    <tm2EnableUserAssignmentLog>true</tm2EnableUserAssignmentLog>
    <t2ForecastAccessLevel>View</t2ForecastAccessLevel>
</Territory2Settings>
```

Usage

Sales Territories components don't support packaging or change sets and aren't supported in CRUD calls.

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

TrailheadSettings

Represents an org's integration with Trailhead for Learning Paths or Enablement programs, including access to enablement sites (formerly myTrailhead).

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

TrailheadSettings values are stored in the `Trailhead.settings` file in the `settings` directory. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

TrailheadSettings components are available in API version 47.0 and later.

Special Access Rules

To access enablement site (myTrailhead) content, the org must have a Sales Enablement license.

Fields

Field Name	Field Type	Description
`enableConfettiEffect`	boolean	Indicates whether animated confetti plays on the screen after a user reaches certain milestones, such as completing an Enablement program in the Guidance Center. The default value of this field is `false`.
`enableMyTrailheadPref`	boolean	Indicates whether the org is connected to an enablement site (myTrailhead). The default value of this field is `true`.
`enableTrailheadInLexTerms`	boolean	Indicates whether the terms and conditions for showing Trailhead content in Lightning Experience are accepted in your org. The default value of this field is `false`.

Declarative Metadata Sample Definition

The following is an example of a TrailheadSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<TrailheadSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <enableMyTrailheadPref>true</enableMyTrailheadPref>
</TrailheadSettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>Trailhead</members>
        <name>Settings</name>
    </types>
    <version>47.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

TrialOrgSettings

Represents the settings in a trial user's org. This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

TrialOrgSettings values are stored in the `TrialOrg.settings` file in the `settings` directory.The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

TrialOrgsettings is available in API version 48.0 and later.

Special Access Rules

Access to TrialOrgSettings requires users to complete the checkout flow in Enterprise, Professional, or Essentials editions. For Essentials, you can also access TrialOrgSettings by completing step 7 of the Setup Assistant.

Fields

Field Name	Field Type	Description
`enableSampleDataDeleted`	boolean	Indicates whether sample data may be deleted on trial orgs (`true`) or not (`false`). The default value is `false`.

Declarative Metadata Sample Definition

The following is an example of a TrialOrgSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<TrialOrgSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <enableSampleDataDeleted>false</enableSampleDataDeleted>
</TrialOrgSettings>
```

Wildcard Support in the Manifest File

This metadata type doesn't support the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

UserEngagementSettings

Represents the metadata associated with various feature settings around Lightning Experience transition and adoption, user engagement and adoption assistance, and adoption apps.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

UserEngagementSettings components have the suffix `.settings` and are stored in the `settings` folder.

Version

Prompt components are available in API version 47.0 and later.

Special Access Rules

See related Salesforce Help for each feature for permission and edition requirements.

Fields

Field Name	Field Type	Description
`canUseAdoptionApps`	boolean	Indicates whether an org can access Lightning Experience transition tools (`true`) or not (`false`). Examples of these tools are Salesforce

Field Name	Field Type	Description
		Optimizer, Lightning Experience Transition Assistant, and the Lightning Experience Readiness Report. The default is `false`. This field applies only to orgs with the External Application Settings page in Setup. Otherwise, this field has no effect. Available in API version 62.0 and later.
`doesScheduledSwitcherRunDaily`	boolean	Indicates where users are automatically switched from Salesforce Classic to Lightning Experience every day (`true`) or weekly (`false`). If `false`, then users are switched weekly. The default is `false`. See Encourage Users to Stay in Lightning Experience in Salesforce Help.
`enableCustomHelpGlobalSection`	boolean	Indicates whether a custom section has been added to the Lightning Experience Help Menu (`true`) or not (`false`). The default is `false`. See Define Custom Help for the Lightning Experience Help Menu in Salesforce Help for more information.
`enableHelpMenuShowFeedback`	boolean	Indicates whether the Give Feedback to Salesforce link in the Lightning Experience Help Menu is visible to users (`true`) or not (`false`). The default is `true`. Even if `false`, admins always see all links in the Help Menu. See Define Custom Help for the Lightning Experience Help Menu in Salesforce Help for more information.
`enableHelpMenuShowHelp`	boolean	Indicates whether the Help For This Page section in the Lightning Experience Help Menu is visible to users (`true`) or not (`false`). The default is `true`. Even if `false`, admins always see all links in the Help Menu. See Define Custom Help for the Lightning Experience Help Menu in Salesforce Help for more information.
`enableHelpMenuShowNewUser`	boolean	Indicates whether the Getting Started section in the Lightning Experience Help Menu is visible to users (`true`) or not (`false`). The default is `true`. Even if `false`, admins always see all links in the Help Menu. See Define Custom Help for the Lightning Experience Help Menu in Salesforce Help for more information.
`enableHelpMenuShowSearch`	boolean	Indicates whether the Search Documentation link in the Lightning Experience Help Menu is visible to users (`true`) or not (`false`). The default is `true`. Even if `false`, admins always see all links in the Help Menu. See Define Custom Help for the Lightning Experience Help Menu in Salesforce Help for more information.
`enableHelpMenuShowSfdcContent`	boolean	Indicates whether any Salesforce-created help resources in Lightning Experience Help Menu are visible to users (`true`) or not (`false`). The default is `true`. Even if `false`, admins always see all links in the Help Menu. See Define Custom Help for the Lightning Experience Help Menu in Salesforce Help for more information.
`enableHelpMenuShowShortcut`	boolean	Indicates whether the View Keyboard Shortcuts link in the Lightning Experience Help Menu is visible to users (`true`) or not (`false`). The default is `true`. Even if `false`, admins always see all links in the Help Menu. See Define Custom Help for the Lightning Experience Help Menu in Salesforce Help for more information.

Field Name	Field Type	Description
enableHelpMenuShowSupport	boolean	Indicates whether the Get Support link in the Lightning Experience Help Menu is visible to users (true) or not (false). The default is true. Even if false, admins always see all links in the Help Menu. See Define Custom Help for the Lightning Experience Help Menu in Salesforce Help for more information.
enableHelpMenuShowTrailhead	boolean	Indicates whether the Go to Trailhead link in the Lightning Experience Help Menu is visible to users (true) or not (false). The default is true. Even if false, admins always see all links in the Help Menu. See Define Custom Help for the Lightning Experience Help Menu in Salesforce Help for more information.
enableIBILOptOutDashboards	boolean	Indicates whether the It's Better in Lightning prompt about Dashboards is hidden from users (true) or not (false). The default is true. Deprecated in API version 51.0 and later.
enableIBILOptOutEvents	boolean	Indicates whether the It's Better in Lightning prompt about Events/Calendar is hidden from users (true) or not (false). The default is true. Deprecated in API version 51.0 and later.
enableIBILOptOutReports	boolean	Indicates whether the It's Better in Lightning prompt about Reports is hidden from users (true) or not (false). The default is true. Deprecated in API version 51.0 and later.
enableIBILOptOutTasks	boolean	Indicates whether the It's Better in Lightning prompt about Tasks is hidden from users (true) or not (false). The default is true. Deprecated in API version 51.0 and later.
enableLexToClassicFeedbackEnable	boolean	Indicates whether the Switch to Salesforce Classic Feedback Form is shown to users (true) or not (false). The default is false. See Switch to Salesforce Classic Feedback Form in Salesforce Help for more information.
enableOrchestrationInSandbox	boolean	Indicates whether adoption assistance and other in-app guidance is shown to users in sandbox orgs (true) or not (false). The default is false. See Define Prompts in Lightning Experience in Salesforce Help for more information.
enableOrgUserAssistEnabled	boolean	Indicates whether all custom in-app guidance created by an org is shown to users (true) or not (false). Doesn't affect active status. The default is true. See Define Prompts in Lightning Experience in Salesforce Help for more information.
enableScheduledSwitcher	boolean	Indicates whether users are automatically switched from Salesforce Classic to Lightning Experience (true) or not (false). The default is true. See Encourage Users to Stay in Lightning Experience in Salesforce Help.
enableSfdcProductFeedbackSurvey	boolean	Indicates whether the Salesforce Product Feedback Form is shown to users (true) or not (false). The default is true. See Salesforce Product Feedback Form in Salesforce Help for more information.

Field Name	Field Type	Description
enableShowSalesforceUserAssist	boolean	Indicates whether all standard in-app guidance created by Salesforce is shown to users (true) or not (false). Doesn't affect active status. The default is true. See Define Prompts in Lightning Experience in Salesforce Help for more information.
isCrucNotificationDisabled	boolean	Indicates whether all notifications about the Winter '20 Turn on Lightning Experience critical update are hidden from admins (true) or not (false). The default is false.
isLEXWelcomeMatDisabled	boolean	Indicates whether the Lightning Experience welcome mat is hidden from users the first time they log into the user interface (true) or not (false). The default is false. See Lightning Experience Welcome Mat in Salesforce Help for more information.
isMeetTheAssistantDisabledInClassic	boolean	Indicates whether all notifications about the Lightning Experience Transition Assistant are hidden from admins in Salesforce Classic (true) or not (false). The default is false.
isMeetTheAssistantDisabledInLightning	boolean	Indicates whether all notifications about the Lightning Experience Transition Assistant are hidden from admins in Lightning Experience (true) or not (false). The default is false.
optimizerAppEnabled	boolean	Indicates whether Salesforce Optimizer is turned on in the org (true) or not (false). The default is false. See Improve Your Implementation with Salesforce Optimizer in Salesforce Help.
suggestedForYou	boolean	Indicates whether Suggested For You is turned on in the org (true) or not (false). The default is true. See Suggested For You in Salesforce Help.

Declarative Metadata Sample Definition

The following is an example of a UserEngagementSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<UserEngagementSettings xmlns="http://soap.sforce.com/2006/04/metadata">
   <canUseAdoptionApps>false</canUseAdoptionApps>
   <doesScheduledSwitcherRunDaily>true</doesScheduledSwitcherRunDaily>
   <enableCustomHelpGlobalSection>true</enableCustomHelpGlobalSection>
   <enableHelpMenuShowFeedback>true</enableHelpMenuShowFeedback>
   <enableHelpMenuShowHelp>true</enableHelpMenuShowHelp>
   <enableHelpMenuShowNewUser>true</enableHelpMenuShowNewUser>
   <enableHelpMenuShowSearch>true</enableHelpMenuShowSearch>
   <enableHelpMenuShowSfdcContent>true</enableHelpMenuShowSfdcContent>
   <enableHelpMenuShowShortcut>true</enableHelpMenuShowShortcut>
   <enableHelpMenuShowSupport>true</enableHelpMenuShowSupport>
   <enableHelpMenuShowTrailhead>true</enableHelpMenuShowTrailhead>
   <enableIBILOptOutDashboards>true</enableIBILOptOutDashboards>
   <enableIBILOptOutEvents>true</enableIBILOptOutEvents>
   <enableIBILOptOutReports>true</enableIBILOptOutReports>
   <enableIBILOptOutTasks>true</enableIBILOptOutTasks>
```

```
    <enableLexToClassicFeedbackEnable>true</enableLexToClassicFeedbackEnable>
    <enableOrgUserAssistEnabled>true</enableOrgUserAssistEnabled>
    <enableScheduledSwitcher>true</enableScheduledSwitcher>
    <enableSfdcProductFeedbackSurvey>true</enableSfdcProductFeedbackSurvey>
    <enableOrchestrationInSandbox>true</enableOrchestrationInSandbox>
    <enableShowSalesforceUserAssist>true</enableShowSalesforceUserAssist>
    <isCrucNotificationDisabled>false</isCrucNotificationDisabled>
    <isLEXWelcomeMatDisabled>false</isLEXWelcomeMatDisabled>
    <isMeetTheAssistantDisabledInClassic>false</isMeetTheAssistantDisabledInClassic>
    <isMeetTheAssistantDisabledInLightning>false</isMeetTheAssistantDisabledInLightning>
    <optimizerAppEnabled>true</optimizerAppEnabled>
</UserEngagementSettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>UserEngagement</members>
        <name>Settings</name>
    </types>
    <version>47.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

UserInterfaceSettings

Represents the settings that modify the behavior of the org's user interface.

Parent Type and Manifest Access

This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

A UserInterfaceSettings component file has the suffix `.settings` and is stored in the `settings` directory. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

UserInterfaceSettings components are available in API version 46.0 and later.

Fields

Field Name	Field Type	Description
dynamicMruActionsOff	boolean	Indicates whether users can create custom actions for their Recently Viewed lists (true) or not (false). The default is true. Available in API version 52.0 and later. Applies to Lightning Experience only.
enableAsyncRelatedLists	boolean	Indicates whether related lists are loaded asynchronously (true) or not (false). The default is false. Available in API version 47.0 and later. Salesforce Classic only.
enableClickjackUserPageHeaderless	boolean	Indicates whether a Visualforce page that hides the standard header has clickjack protections (true) or not (false). The default is true. This setting applies to all of your Visualforce pages.
enableCollapsibleSections	boolean	Indicates whether users are allowed to collapse or expand sections in record details by using the arrow icon next to the section heading. The default is true.
enableCollapsibleSidebar	boolean	Indicates whether users are allowed to show or hide the sidebar on every page that normally includes it (true) or not (false). The default is false. Applies to Salesforce Classic only.
enableCustomObjectTruncate	boolean	Indicates whether users with Customize Application permission can truncate custom objects (true) or not (false). When you truncate an object, you delete the object's associated records permanently, while preserving the empty object and its metadata. The default is false. Available in API version 47.0 and later.
enableCustomSidebarOnAllPages	boolean	Indicates whether custom sidebar components are available on all pages for all org users (true) or not (false). The default is false. Applies to Salesforce Classic only.
enableDeleteFieldHistory	boolean	Indicates whether users can delete field history and field history archive records (true) or not (false). The default is false. Available in API version 47.0 and later.
enableExternalObjectAsyncRelatedLists	boolean	Indicates whether related lists of external objects are loaded asynchronously (true) or not (false). The default is true. Available in API version 48.0 and later. Salesforce Classic only.
enableHoverDetails	boolean	Indicates whether an interactive overlay containing record details is displayed (true) or not (false). The default is true. Note: To view hover details for a record, users need the appropriate sharing access and field-level security access for the fields in the mini page layout.
enableInlineEdit	boolean	Indicates whether users are allowed to edit field values on a record's detail page (true) or not (false). The default is true.

Field Name	Field Type	Description
enablePersonalCanvas	boolean	Indicates whether users can install and use personal canvas apps (true) or not (false). The default is true. This setting applies to all of your Visualforce pages.
enableRelatedListHovers	boolean	Indicates whether related list hover links display at the top of record detail pages and custom object detail pages in Setup (true) or not (false). Users can hover over a related list link to display the list and its number of records in an interactive overlay. Users quickly view and manage the related list items from the overlay. Users can also click a related list hover link to jump to the related list without having to scroll down the page. The default is true. Available in API version 50.0 and later.
enableQuickCreate	boolean	Indicates whether an area displays on a tab home page (corresponds to the **Show Quick Create** setting), allowing users to create a record quickly with minimal information (true) or not (false). The Quick Create area displays by default on the tab home pages for leads, accounts, contacts, and opportunities. You can control whether the Quick Create area is displayed on all relevant tab home pages.

Declarative Metadata Sample Definition

The following is an example of a UserInterfaceSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<UserInterfaceSettings xmlns="http://soap.sforce.com/2006/04/metadata">
  <enableDeleteFieldHistory>false</enableDeleteFieldHistory>
  <enableInlineEdit>true</enableInlineEdit>
  <enableHoverDetails>false</enableHoverDetails>
  <enableQuickCreate>true</enableQuickCreate>
  <enablePersonalCanvas>false</enablePersonalCanvas>
  <enableClickjackUserPageHeaderless>true</enableClickjackUserPageHeaderless>
</UserInterfaceSettings>
```

Example Package Manifest

The following is an example package manifest used to deploy or retrieve the user interface settings metadata for an organization:

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
 <types>
  <members>UserInterface</members>
  <name>Settings</name>
 </types>
 <version>46.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

UserManagementSettings

Represents a selection of user management options that appear on the User Management Settings Setup page. This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

UserMangementSettings are stored in the `UserManagement.settings` directory. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

Manage org-wide settings for certain options. User Management Settings are available in API version 46.0 and later.

Fields

Field	Field Type	Description
`enableConcealPersonalInfo`	boolean	Indicates if personal information fields in user records are hidden from external users (`true`) or not (`false`). When this field is set to `true`, 10 personal information fields are hidden. The default value is `false`. This field is unavailable for orgs created in Winter '22 or later. Salesforce recommends that you use the `enableEnhancedConcealPersonalInfo` field instead of `enableConcealPersonalInfo`. Before you set the `enableEnhancedConcealPersonalInfo` field to `true`, make sure that `enableConcealPersonalInfo` is set to `false`.
`enableContactlessExternalIdentityUsers`	boolean	If `true` and your org has the External Identity license, you can create contactless users. Having users without contact information reduces the overhead of managing customers. Purchase the External Identity license to access the Customer 360 Identity product. The default is `false`. Available in API version 47.0 and later.

Field	Field Type	Description
enableEnhancedConcealPersonalInfo	boolean	Indicates if personal information fields in user records are hidden from external users (true) or not (false). When this field is set to true, you can choose which fields are classified as personal information and hidden on the User Management Settings Setup page. The default value is false. This field is available in API version 53.0 and later. Before you set the enableEnhancedConcealPersonalInfo field to true, make sure that enableConcealPersonalInfo is set to false.
enableEnhancedPermsetMgmt	boolean	If you enable **Enhanced Permission Set Component Views** (true), you can work with permission sets more easily. For example, when you have large numbers of Apex class assignments for permission sets, you can enable a paginated result set, standard filtering, and sorting.
enableEnhancedProfileMgmt	boolean	If you enable **Enhanced Profile Lists Views** (true), you can quickly view, customize, and edit list data.
enableEnhcUiUserAccessPolicies	boolean	Indicates whether you create and manage user access policies through an improved user interface (true) or not (false). The default value is false. If user access policies aren't enabled, this field has no effect. If user access policies are enabled, this field is automatically set to true, but you can change it to false. Available in API version 60.0 and later.
enableNewProfileUI	boolean	If you enable **Enhanced Profile User Interface** (true), you can use the streamlined, enhanced profile user interface to browse, search, and modify settings. You can use only one user interface at a time.
enableProfileFiltering	boolean	With profile filtering enabled (true), you can restrict who sees profile names to the users who require the access for their job roles. If profile filtering is disabled (false), users can see all profiles in a Salesforce org, regardless of which permissions they have. Important: Profile names are also exposed when users with permissions to perform the following tasks take these actions: • Create a tab or record type with a wizard step that includes the assignment of tabs and record types to profiles.

Field	Field Type	Description
		• Configure a login flow where viewing profile lists is required to make flow associations. • Set up delegated admins where looking up profiles is needed to identify assignable profiles. • Administer an org as a delegated customer admin. • Administer an org as a delegated admin to view and assign profiles of the delegated group. This field is available in API version 50.0 and later.
`enableRestrictEmailDomains`	boolean	Indicates whether the Email Domain Allowlist is visible (`true`) or hidden (`false`) in Setup. The default value is `false`. This field is available in API version 53.0 and later.
`enableScrambleUserData`	boolean	If you enable **Let Users Scramble Their User Data** (`true`), users can request that Salesforce remove all their personal data. Because Salesforce can't delete information, it scrambles their data. Scrambling a user's data is unrecoverable. So this org-wide setting serves as an extra precaution. If a user requests it, you scramble the data programmatically with the `obfuscateUser` Apex method. You can use the method, for example, in a custom Apex trigger, workflow, or the Developer Console. This field is available in API version 47.0 and later.
`enableUserSelfDeactivate`	boolean	If you enable **User Self Deactivate** (`true`), users can deactivate their Experience Cloud site or Chatter accounts.
`enhancedPermSetList`	boolean	Indicates whether you manage permission sets with an updated user interface on the Permissions Setup page (`true`) or not (`false`). The default value is `false`. Available in API version 63.0 and later.
`enhancedUserListView`	boolean	Indicates whether you manage users with an updated user interface on the Users Setup page (`true`) or not (`false`). The default value is `false`. Available in API version 62.0 and later.
`enhancedUserRoleListView`	boolean	Indicates whether you manage roles with an updated user interface on the Roles Setup page (`true`) or not

Field	Field Type	Description
		(`false`). The default value is `false`. Available in API version 63.0 and later.
`groupSummaryUIEnhancement`	boolean	Indicates whether you use an improved user interface to add or remove public group members through the group's summary page `true` or not (`false`). If this field is set to `true`, you can still manage public group membership through the group's detail page. The default value is `true`. Available in API version 62.0 and later.
`permsetsInFieldCreation`	boolean	If `true`, users can assign field-level security to permission sets instead of to profiles when creating a field on an object, setting field-level security on a field, or changing a field type on a field. The default is `false`. Available in API version 56.0 and later.
`psaExpirationUIEnabled`	boolean	Indicates if admins can use an updated user interface that includes an assignment expiration for permission sets and permission set groups (`true`) or not (`false`). The default value is `false`. This field is available in API version 52.0 and later.
`restrictedProfileCloning`	boolean	When enabled (`true`), only permissions accessible to your org are enabled when you clone profiles. When disabled (`false`), all permissions currently enabled in the source profile are also enabled for the cloned profile, even if your org can't currently access them. This field is available in API version 50.0 and later.
`userAccessPoliciesEnabled`	boolean	Indicates if user access policies are enabled (`true`) or not (`false`). With user access policies, you can automate and migrate your users' assignments to managed package licenses, permission sets, and other access mechanisms based on criteria that you set. The default value is `false`. This field is available in API version 58.0 and later.

Declarative Metadata Sample Definition

The following is an example of a UserManagementSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<UserManagementSettings xmlns="http://soap.sforce.com/2006/04/metadata"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <enableConcealPersonalInfo>false</enableConcealPersonalInfo>
   <enableContactlessExternalIdentityUsers>false</enableContactlessExternalIdentityUsers>

    <enableEnhancedConcealPersonalInfo>true</enableEnhancedConcealPersonalInfo>
```

```
    <enableEnhancedPermsetMgmt>false</enableEnhancedPermsetMgmt>
    <enableEnhancedProfileMgmt>true</enableEnhancedProfileMgmt>
    <enableNewProfileUI>false</enableNewProfileUI>
    <enableProfileFiltering>false</enableProfileFiltering>
    <enableRestrictEmailDomains>true</enableRestrictEmailDomains>
    <enableScrambleUserData>false</enableScrambleUserData>
    <enableUserSelfDeactivate>false</enableUserSelfDeactivate>
    <enhancedPermSetList>true</enhancedPermSetList>
    <enhancedUserListView>true</enhancedUserListView>
    <enhancedUserRoleListView>true</enhancedUserRoleListView>
    <groupSummaryUIEnhancement>true</groupSummaryUIEnhancement>
    <restrictedProfileCloning>true</restrictedProfileCloning>
    <userAccessPoliciesEnabled>true</userAccessPoliciesEnabled>
</UserManagementSettings>
```

The following is an example `package.xml` manifest that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>UserManagement</members>
        <name>Settings</name>
    </types>
    <version>53.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

VoiceSettings

Represents an org's Sales Dialer settings, such as call recording, conferencing, and voicemail.

This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for details.

File Suffix and Directory Location

VoiceSettings values are stored in the `Voice.settings` file in the `settings` directory. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

VoiceSettings is available in API version 47.0 and later.

Fields

Field Name	Field Type	Description
`enableCallDisposition`	boolean	Indicates whether call disposition is enabled (`true`) or not (`false`). With call disposition, also called Call Result, sales reps can track whether a call was connected and how it went. Default value is `false`. To use this feature, enable Dialer in Lightning Experience.
`enableConsentReminder`	boolean	Indicates whether the consent reminder is enabled (`true`) or not (`false`). With the consent reminder, prior to recording a call, users see a prompt reminding them not to record phone calls without consent. Default value is `false`. To use this feature, enable Dialer in Lightning Experience.
enableDefaultRecording	boolean	Indicates whether the default recording is enabled (`true`) or not (`false`). With default recording, sales reps can record calls automatically in the Sales Dialer. Default value is `false`. This field is available in API version 54.0 and later.
`enableVoiceCallList`	boolean	Indicates whether Call List is enabled (`true`) or not (`false`). Sales reps can use call list to keep a running list of the calls they want to make. Default value is `false`. To use this feature, enable Dialer in Lightning Experience.
`enableVoiceCallRecording`	boolean	Indicates whether Call Recording is enabled (`true`) or not (`false`). Sales reps can record important calls directly from the call panel in Sales Dialer. Default value is `false`. To use this feature, enable Dialer in Lightning Experience.
`enableVoiceCoaching`	boolean	Indicates whether Call Monitoring is enabled (`true`) or not (`false`). Using the Monitor tab in the call panel, managers can listen to the calls of their sales reps for personalized coaching. Default value is `false`. To use this feature, enable Dialer in Lightning Experience.
`enableVoiceConferencing`	boolean	Reserved for future use.
`enableVoiceLocalPresence`	boolean	Indicates whether Local Presence is enabled (`true`) or not (`false`). Local Presence displays phone numbers with the same area code as the prospects your reps are calling, so more calls are answered. Default value is `false`. To use this feature, enable Dialer in Lightning Experience.

Field Name	Field Type	Description
`enableVoiceMail`	boolean	Indicates whether voicemail is enabled (`true`) or not (`false`). Sales reps can receive and store up to 20 personal voicemail messages in Salesforce. Default value is `false`. To use this feature, enable Dialer in Lightning Experience.
`enableVoiceMailDrop`	boolean	Indicates whether Voicemail Drop is enabled (`true`) or not (`false`). Sales reps can "drop" (or send) prerecorded messages to recipients' voicemail boxes. Default value is `false`. To use this feature, enable Dialer in Lightning Experience.

Declarative Metadata Sample Definition

The following is an example of the package file.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
    </types>
    <version>28.0</version>
</Package>
```

The package file references the following Voice.settings file.

```
<?xml version="1.0" encoding="UTF-8"?>
<VoiceSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <enableCallDisposition>true</enableCallDisposition>
    <enableVoiceCallList>true</enableVoiceCallList>
    <enableVoiceCallRecording>true</enableVoiceCallRecording>
    <enableVoiceCoaching>true</enableVoiceCoaching>
    <enableVoiceConferencing>true</enableVoiceConferencing>
    <enableVoiceLocalPresence>true</enableVoiceLocalPresence>
    <enableVoiceMail>true</enableVoiceMail>
    <enableVoiceMailDrop>true</enableVoiceMailDrop>
</VoiceSettings>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

WarrantyLifeCycleMgmtSettings

Represents settings that control the Warranty Administration for your org.

This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for more details.

File Suffix and Directory Location

WarrantyLifeCycleMgmtSettings values are stored in the `WarrantyLifecycleMgmt.settings` file in the `settings` directory.

Version

WarrantyLifeCycleMgmtSettings components are available in API version 54.0 and later.

Fields

Field Name	Field Type	Description
`enableWarrantyLCMgmt`	boolean	Indicates whether warranty life-cycle management is enabled in your org (`true`) or not `false`).

Declarative Metadata Sample Definition

The following is an example of WarrantyLifeCycleMgmtSettings component.

```
<!--
   ~ Copyright 2022 salesforce.com, inc.
   ~ All Rights Reserved
   ~ Company Confidential
   -->
<WarrantyLifecycleMgmtSettings
 xmlns="http://soap.sforce.com/2006/04/metadata">
 <enableWarrantyLCMgmt>true</enableWarrantyLCMgmt>
</WarrantyLifecycleMgmtSettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
   ~ Copyright 2022 salesforce.com, inc.
   ~ All Rights Reserved
   ~ Company Confidential
   -->
<Package
 xmlns="http://soap.sforce.com/2006/04/metadata">
 <types>
  <members>WarrantyLifecycleMgmt</members>
  <name>Settings</name>
 </types>
 <version>54.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

WorkDotComSettings

Represents WorkDotCom settings. This type extends the Metadata metadata type and inherits its `fullName` field.

Version

WorkDotComSettings components are available in API version 31.0 and later.

Fields

Field Name	Field Type	Description
`enableCoachingManagerGroupAccess`	boolean	Indicates whether Coaching Manager Group Access is available to users `(true)` or not `(false)`. Default value is `true`. Deprecated.
`enableGoalManagerGroupAccess`	boolean	Indicates whether Goal Manager Group Access is available to users `(true)` or not `(false)`. Default value is `true`. Deprecated.
`enableProfileSkills`	boolean	Indicates whether Profile Skills is available to users `(true)` or not `(false)`. Default value is `true`.
`enableProfileSkillsAddFeedPost`	boolean	Indicates whether Add Skills as Feed Posts is available to users `(true)` or not `(false)`. Default value is `true`.
`enableProfileSkillsAutoSuggest`	boolean	Indicates whether Profile Skills Auto Suggest is available to users `(true)` or not `(false)`. Default value is `true`.
`enableProfileSkillsUsePlatform`	boolean	Indicates whether Profile Skills Use Platform is available to users `(true)` or not `(false)`. Default value is `true`.
`enableWorkBadgeDefRestrictPref`	boolean	Indicates whether Badge Definition Restriction is available to users `(true)` or not `(false)`. Default value is `true`. Deprecated.
`enableWorkCalibration`	boolean	Indicates whether Calibration is available to users `(true)` or not `(false)`. Default value is `false`. Deprecated.
`enableWorkCanvasPref`	boolean	Indicates whether Canvas is available to users `(true)` or not `(false)`. Default value is `true`. Deprecated.

Field Name	Field Type	Description
enableWorkCertification	boolean	Indicates whether Certification is available to users (true) or not (false). Default value is true. Deprecated.
enableWorkCertificationNotification	boolean	Indicates whether Certification Notification is available to users (true) or not (false). Default value is false. Deprecated.
enableWorkRewardsPref	boolean	Indicates whether Rewards is available to users (true) or not (false). Default value is true. Deprecated.
enableWorkThanksPref	boolean	Indicates whether Thanks is available to users (true) or not (false). Default value is true.

Declarative Metadata Sample Definition

The following is an example of a WorkDotComSettings component.

```
<?xml version="1.0" encoding="UTF-8"?>
<WorkDotComSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <enableCoachingManagerGroupAccess>true</enableCoachingManagerGroupAccess>
    <enableGoalManagerGroupAccess>true</enableGoalManagerGroupAccess>
    <enableProfileSkills>true</enableProfileSkills>
    <enableProfileSkillsAddFeedPost>true</enableProfileSkillsAddFeedPost>
    <enableProfileSkillsAutoSuggest>true</enableProfileSkillsAutoSuggest>
    <enableProfileSkillsUsePlatform>true</enableProfileSkillsUsePlatform>
    <enableWorkBadgeDefRestrictPref>true</enableWorkBadgeDefRestrictPref>
    <enableWorkCalibration>true</enableWorkCalibration>
    <enableWorkCanvasPref>true</enableWorkCanvasPref>
    <enableWorkCertification>true</enableWorkCertification>
    <enableWorkCertificationNotification>true</enableWorkCertificationNotification>
    <enableWorkRewardsPref>true</enableWorkRewardsPref>
    <enableWorkThanksPref>true</enableWorkThanksPref>
    </WorkDotComSettings>
```

WorkforceEngagementSettings

Represents settings for Workforce Engagement Management.

File Suffix and Directory Location

WorkforceEngagementSettings components are stored in the WorkforceEngagement.settings folder.

Version

WorkforceEngagementSettings is available in API version 52.0 and later.

Special Access Rules

To use Workforce Engagement settings, the org requires a Workforce Engagement license.

Fields

Field Type

Field Name	Field Type	Description
enableMachineLearningForecasting	boolean	Indicates whether machine learning-based forecasting is used (`true`) or not used (`false`).
enableWorkforceEngagement	boolean	Indicates whether Workforce Engagement is enabled (`true`) or not enabled (`false`).
enableWorkforceEngagementConfiguration	boolean	Indicates whether the Workforce Engagement Configuration App is installed or enabled (`true`) or not (`false`). If `true`, it grants access to the Lightning App as well as the app's Job Profile Mapping tab. It also defaults the standard and custom profile tab settings to On. If `false`, it removes access to the app and tab but doesn't delete the app metadata. This field is available in API version 53.0 and later.
enableHistoricalAdherence	boolean	Indicates whether historical adherence is enabled (`true`) or not enabled (`false`). This field is available in API version 54.0 and later.
enableIndividualAdherence	boolean	Indicates whether individual adherence is enabled (`true`) or not enabled (`false`). This field is available in API version 54.0 and later.
enableIntradayManagement	boolean	Indicates whether the intraday management dashboard is enabled (`true`) or not enabled (`false`). This field is available in API version 55.0 and later.
enableRealTimeAdherence	boolean	Indicates whether real-time adherence is enabled (`true`) or not enabled (`false`). To use real-time adherence, you must also enable Omni-Channel. This field is available in API version 55.0 and later.

Declarative Metadata Sample Definition

The following is an example of a `WorkforceEngagement.settings` component.

```
<?xml version="1.0" encoding="UTF-8"?>
<WorkforceEngagementSettings xmlns="http://soap.sforce.com/2006/04/metadata">
   <enableMachineLearningForecasting>true</enableMachineLearningForecasting>
   <enableWorkforceEngagement>true</enableWorkforceEngagement>
   <enableWorkforceEngagementConfiguration>true</enableWorkforceEngagementConfiguration>
   <enableHistoricalAdherence>true</enableHistoricalAdherence>
   <enableenableIndividualAdherence>true</enableIndividualAdherence>
   <enableIntradayManagement>true</enableIntradayManagement>
   <enableRealTimeAdherence>true</enableRealTimeAdherence>
</WorkforceEngagementSettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>WorkforceEngagement</members>
        <name>Settings</name>
    </types>
    <55.0>[ftest]</55.0>
</Package>
```

Usage

When enableMachineLearningForecasting is set to `false`, we clean up data from our Ofek forecasting platform. The original copy of the same set of data is stored in the Core app, so no data is lost.

SharedTo

SharedTo defines the sharing access for a list view or a folder. It can be used to specify the target and source for owner-based sharing rules.

See Sharing Considerations and Public and Personal Groups in Salesforce Help.

Note: SharedTo on page 2150 is included in the metadata for shared and private list views. SharedTo on page 2150 isn't in the metadata for public list views.

Declarative Metadata File Suffix and Directory Location

SharedTo on page 2150 is used with ListView, Folder, and SharingRules.

Version

SharedTo on page 2150 is available in API version 17.0 and later.

Fields

Field	Field Type	Description
`allCustomerPortalUsers`	string	A group containing all customer portal users. This field is available in API version 24.0 and later.
`allInternalUsers`	string	A group containing all internal and nonportal users. This field is available in API version 24.0 and later.
`allPartnerUsers`	string	A group containing all partner users. This field is available in API version 24.0 and later.

Field	Field Type	Description
channelProgramGroup	string	A system-managed group with sharing access containing all partner members of the corresponding channel program or level. This field is available in API version 41.0 and later.
channelProgramGroups	string[]	A list of system-managed groups with sharing access containing all partner members of the corresponding channel programs or levels. This field is available in API version 41.0 and later.
group	string[]	A list of groups with sharing access. Use this field instead of the groups field. This field is available in API version 22.0 and later.
guestUser	string[]	A list of guest user nicknames with sharing access. This field can be used only with SharingGuestRule. This field is available in API version 47.0 and later.
groups	string[]	A list of groups with sharing access. Use the group field instead for API version 22.0 and later.
managerSubordinates	string[]	A list of users whose direct and indirect subordinates receive sharing access. This field is available in API version 24.0 and later.
managers	string[]	A list of users whose direct and indirect managers receive sharing access. This field is available in API version 24.0 and later.
portalRole	string[]	A list of groups with sharing access containing all users in a portal role. This field is available in API version 24.0 and later.
portalRoleandSubordinates	string[]	A list of groups with sharing access containing all users in a portal role or any users under that role. This field is available in API version 24.0 and later.
role	string[]	A list of roles with sharing access. Use this field instead of the roles field. This field is available in API version 22.0 and later.
roleAndSubordinates	string[]	A list of roles with sharing access. All roles below each of these roles in the role hierarchy also have sharing access. If portal accounts are enabled, then all roles and portal accounts below each of these roles in the role hierarchy also have sharing access. Use this field instead of the rolesAndSubordinates field.

Field	Field Type	Description
		This field is available in API version 22.0 and later. The availability of this field depends on the type of org, org creation date, release update enforcement status, and whether digital experiences is enabled. For more information, see this knowledge article.
`roleAndSubordinatesInternal`	string[]	A list of roles with sharing access. All roles below each of these roles in the role hierarchy also have sharing access. This field is available in API version 22.0 and later. The availability of this field depends on the type of org, org creation date, release update enforcement status, and whether digital experiences is enabled. For more information, see this knowledge article.
`roles`	string[]	A list of roles with sharing access. Use the `role` field instead for API version 22.0 and later.
`rolesAndSubordinates`	string[]	A list of roles with sharing access. All roles below each of these roles in the role hierarchy also have sharing access. If portal accounts are enabled, then all roles and portal accounts below each of these roles in the role hierarchy also have sharing access. Use the `roleAndSubordinates` field instead for API version 22.0 and later.
`territories`	string[]	A list of territories with sharing access. Use the `territory` field instead for API version 22.0 and later.
`territoriesAndSubordinates`	string[]	A list of territories with sharing access. All territories below each of these territories in the territory hierarchy also have sharing access. Use the `territoryAndSubordinates` field instead for API version 22.0 and later.
`territory`	string[]	A list of territories with sharing access. Use this field instead of the `territories` field. If you're using Sales Territories, use *`modelName.territoryName`* for the shared-to and shared-from `territory` values, where: • *`modelName`* equals the name of the active territory model in the API. • *`territoryName`* equals the territory's name in the API. This field is available in API version 22.0 and later.
`territoryAndSubordinates`	string[]	A list of territories with sharing access. All territories below each of these territories in the territory hierarchy also have sharing

Field	Field Type	Description
		access. Use this field instead of the `territoriesAndSubordinates` field. If you're using Sales Territories, use *`modelName.territoryName`* for the shared-to and shared-from `territoryAndSubordinates` values, where: • *`modelName`* equals the name of the active territory model in the API. • *`territoryName`* equals the territory's name in the API. This field is available in API version 22.0 and later.
`queue`	string[]	A list of queues with sharing access. Applies only to lead, case, and CustomObject sharing rules. This field is available in API version 24.0 and later.

SharingBaseRule

Represents sharing rule settings such as access level and to whom access is granted.

This type extends the Metadata metadata type and inherits its `fullName` field.

Note: You can't create a SharingBaseRule on page 2153 component directly. Use the components under SharingRules instead.

Version

SharingBaseRule on page 2153 replaces BaseSharingRule and is available in API version 33.0 and later.

Fields

Field	Field Type	Description
`accessLevel`	string	Required. The access level that the sharing rule grants.
`accountSettings`	AccountSharingRuleSettings[]	The access level for the account's children (case, contact, and opportunity).
`description`	string	Describes the sharing rule. Maximum of 1000 characters.
`label`	string	Required. Label for the sharing rule.
`sharedTo`	SharedTo	Required. Specifies who the record is shared with.

AccountSharingRuleSettings

Defines the access level for the case, contact, and opportunity associated with the account.

Field	Field Type	Description
`caseAccessLevel`	string	Required. The access level that the user or group has to cases associated with the account. Possible values are: • None • Read • Edit
`contactAccessLevel`	string	Required. The access level that the user or group has to contacts associated with the account. Possible values are: • None • Read • Edit
`opportunityAccessLevel`	string	Required. The access level that the user or group has to opportunities associated with the account. Possible values are: • None • Read • Edit

Wildcard Support in the Manifest File

This metadata type doesn't support the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SharingRules

Represents the base container for sharing rules, which can be criteria-based, ownership-based, territory-based, or for guest user access. SharingRules enables you to share records with a set of users, using rules that specify the access level for the target user group.

This type extends the Metadata metadata type and inherits its `fullName` field. For more information, see "Sharing Rules" in Salesforce Help.

In API version 33.0 and later, retrieving, deleting, or deploying of all sharing rules in an organization is available. Wildcard support is also available. You can't retrieve, delete, or deploy manual sharing rules or sharing rules by their type (owner, criteria-based, territory, or guest user).

Declarative Metadata File Suffix and Directory Location

In API version 33.0 and later, components are stored in the sharingRules folder and their file name matches the object name with the suffix `.sharingRules`. Criteria-based, owner-based, territory-based, and guest user sharing rules are all contained in a `object.sharingRule` file.

Before API version 33.0, SharingRules components are stored in their corresponding object directory and the file name matches the object name. For example, the `accountSharingRules` directory contains an `Account.sharingRules` file for account sharing rules. SharingRules for custom objects are stored in the `customObjectSharingRules` directory, which contains files with the `.sharingRules` extension such as `ObjA__c.sharingRules`, where ObjA refers to the developer name of a custom object type.

Version

SharingRules components are available in API version 24.0 and later, but these components are no longer available in API version 33.0 and later: AccountSharingRules, CampaignSharingRules, CaseSharingRules, ContactSharingRules, LeadSharingRules, OpportunitySharingRules, AccountTerritorySharingRules, CustomObjectSharingRules, UserSharingRules.

In API version 33.0 and later, use SharingCriteriaRule, SharingOwnerRule and SharingTerritoryRule.

Special Access Rules

As of Spring '20 and later, only users with the View Setup and Configuration permission can access this object, and only users with the Manage Sharing permission can edit this object.

Fields

The following information assumes that you're familiar with implementing sharing rules for standard objects and custom objects. For more information on these fields, see "Sharing Settings" in Salesforce Help.

Field	Field Type	Description
`sharingCriteriaRules`	SharingCriteriaRule[]	An array of criteria-based sharing rules. Available in API version 33.0 and later.
`sharingGuestRules`	SharingGuestRule[]	An array of guest user sharing rules. Available in API version 47.0 and later.
`sharingOwnerRules`	SharingOwnerRule[]	An array of ownership-based sharing rules. Available in API version 33.0 and later.
`sharingTerritoryRules`	SharingTerritoryRule[]	An array of territory-based sharing rules. Available in API version 33.0 and later.

SharingCriteriaRule

Defines a criteria-based sharing rule. It extends SharingBaseRule and inherits all its fields. Available in API version 33.0 and later.

Field	Field Type	Description
booleanFilter	string	Advanced filter conditions that are specified for the sharing rule.
criteriaItems	FilterItem[]	An array of the boolean criteria (conditions) for the sharing rule.
includeRecordsOwnedByAll	boolean	Required. Indicates whether records owned by users who can't have an assigned role are included in the records shared (`true`) or not (`false`). Examples of users who can't have an assigned role are high-volume users and system users such as automated process users.or Salesforce system users. You can't edit this field after the sharing rule is created.

SharingGuestRule

Defines a guest user sharing rule. It extends SharingBaseRule and inherits all its fields, except `accountSettings`. Available in API version 47.0 and later.

Note: For SharingGuestRule, the `accessLevel` field can be set only to `Read`.

Field	Field Type	Description
booleanFilter	string	Advanced filter conditions that are specified for the sharing rule. Available in API version 48.0 and later.
criteriaItems	FilterItem[]	An array of the boolean criteria (conditions) for the sharing rule. Available in API version 48.0 and later.
includeHVUOwnedRecords	boolean	Required. Indicates whether records owned by high-volume community or site users are included in the records shared (`true`) or not (`false`). By default, only records owned by authenticated users, guest users, and queues are included in sharing rules. This field has a default value of `false`. Available in API version 52.0 and later. You can't edit this field after the sharing rule is created.

SharingOwnerRule

Defines an ownership-based sharing rule. It extends SharingBaseRule and inherits all its fields. Available in API version 33.0 and later.

Field	Field Type	Description
sharedFrom	SharedTo	Required. Specifies the record owners. If you're using Sales Territories, use *`modelName.territoryName`* for the shared-to

Field	Field Type	Description
		and shared-from `territory` and `territoryAndSubordinates` values on the SharedTo type, where: • *`modelName`* equals the name of the active territory model in the API. • *`territoryName`* equals the territory's name in the API.

SharingTerritoryRule

Defines a territory-based sharing rule. It extends SharingOwnerRule and inherits all its fields. Available in API version 33.0 and later.

AccountSharingRules

Represents the sharing rules for accounts. It extends the SharingRules metadata type and inherits its `fullName` field. Only available in API version 32.0 and earlier.

Field	Field Type	Description
`criteriaBasedRules`	AccountCriteriaBasedSharingRule[]	List that defines user criteria-based rules.
`ownerRules`	AccountOwnerSharingRule[]	List that defines user membership-based rules.

CampaignSharingRules

Represents the sharing rules for campaigns. It extends the SharingRules metadata type and inherits its `fullName` field. Only available in API version 32.0 and earlier.

Field	Field Type	Description
`criteriaBasedRules`	CampaignCriteriaBasedSharingRule[]	List that defines user criteria-based rules.
`ownerRules`	CampaignOwnerSharingRule[]	List that defines user membership-based rules.

CaseSharingRules

Represents the sharing rules for cases. It extends the SharingRules metadata type and inherits its `fullName` field. Only available in API version 32.0 and earlier.

Field	Field Type	Description
`criteriaBasedRules`	CaseCriteriaBasedSharingRule[]	List that defines user criteria-based rules.
`ownerRules`	CaseOwnerSharingRule[]	List that defines user membership-based rules.

ContactSharingRules

Represents the sharing rules for contacts. It extends the SharingRules metadata type and inherits its `fullName` field. Only available in API version 32.0 and earlier.

Field	Field Type	Description
`criteriaBasedRules`	ContactCriteriaBasedSharingRule[]	List that defines user criteria-based rules.
`ownerRules`	ContactOwnerSharingRule[]	List that defines user membership-based rules.

LeadSharingRules

Represents the sharing rules for leads. It extends the SharingRules metadata type and inherits its `fullName` field. Only available in API version 32.0 and earlier.

Field	Field Type	Description
`criteriaBasedRules`	LeadCriteriaBasedSharingRule[]	List that defines user criteria-based rules.
`ownerRules`	LeadOwnerSharingRule[]	List that defines user membership-based rules.

OpportunitySharingRules

Represents the sharing rules for opportunities. It extends the SharingRules metadata type and inherits its `fullName` field. Only available in API version 32.0 and earlier.

Field	Field Type	Description
`criteriaBasedRules`	OpportunityCriteriaBasedSharingRule[]	List that defines user criteria-based rules.
`ownerRules`	OpportunityOwnerSharingRule[]	List that defines user membership-based rules.

AccountTerritorySharingRules

Represents the sharing rules for account territories in the original territory management feature. It extends the SharingRules metadata type and inherits its `fullName` field. Only available in API version 32.0 and earlier.

Field	Field Type	Description
`rules`	AccountTerritorySharingRule[]	List that defines user membership-based rules. The list of acceptable values for the `sharedFrom` fields are: • `territory` • `territoryAndSubordinates`

CustomObjectSharingRules

Represents the sharing rules for custom objects. It extends the SharingRules metadata type and inherits its `fullName` field. Only available in API version 32.0 and earlier.

Field	Field Type	Description
`criteriaBasedRules`	CustomObjectCriteriaBasedSharingRule[]	List that defines user criteria-based rules.
`ownerRules`	CustomObjectOwnerSharingRule[]	List that defines user membership-based rules.

UserSharingRules

Represents the sharing rules for users. With user sharing rules, you can share members of a group with members of another group. It extends the SharingRules metadata type and inherits its `fullName` field. Only available in API version 32.0 and earlier.

Field	Field Type	Description
`criteriaBasedRules`	UserCriteriaBasedSharingRule[]	List that defines user criteria-based rules.
`membershipRules`	UserMembershipSharingRule[]	List that defines user membership-based rules.

Declarative Metadata Sample Definition

For retrieving sharing rules, see `package.xml` sample at SharingRules.

The following sample XML definition represents a criteria-based sharing rule in API version 33.0.

```
<?xml version="1.0" encoding="UTF-8"?>
<SharingRules xmlns="http://soap.sforce.com/2006/04/metadata">
    <sharingCriteriaRules>
        <fullName>AccountCriteriaShareWithCEO</fullName>
        <accessLevel>Edit</accessLevel>
        <accountSettings>
            <caseAccessLevel>Read</caseAccessLevel>
            <contactAccessLevel>Edit</contactAccessLevel>
            <opportunityAccessLevel>Edit</opportunityAccessLevel>
        </accountSettings>
        <criteriaItems>
            <field>Name</field>
            <operation>startsWith</operation>
            <value>Test</value>
        </criteriaItems>
        <description>my account criteria rule description</description>
        <label>AccountCriteriaShareWithCEO</label>
        <sharedTo>
            <role>CEO</role>
        </sharedTo>
    </sharingCriteriaRules>
</SharingRules>
```

The following sample XML definition represents an ownership-based sharing rule in API version 33.0.

```
<?xml version="1.0" encoding="UTF-8"?>
<SharingRules xmlns="http://soap.sforce.com/2006/04/metadata">
    <sharingOwnerRules>
        <fullName>MyCase</fullName>
        <accessLevel>Edit</accessLevel>
        <description>my case test owner sharing rule desc</description>
        <label>MyCase</label>
        <sharedFrom>
            <role>COO</role>
        </sharedFrom>
        <sharedTo>
            <role>CEO</role>
        </sharedTo>
    </sharingOwnerRules>
</SharingRules>
```

The following sample XML definition represents a territory-based sharing rule in API version 33.0.

```
<?xml version="1.0" encoding="UTF-8"?>
<SharingRules xmlns="http://soap.sforce.com/2006/04/metadata">
    <sharingTerritoryRules>
        <fullName>MyAccountTerritoryRule</fullName>
        <accessLevel>Read</accessLevel>
        <accountSettings>
            <caseAccessLevel>None</caseAccessLevel>
            <contactAccessLevel>Read</contactAccessLevel>
            <opportunityAccessLevel>None</opportunityAccessLevel>
        </accountSettings>
        <description>MyAccountTerritoryRule desc</description>
        <label>MyAccountTerritoryRule</label>
        <sharedFrom>
            <territory>My_territory</territory>
        </sharedFrom>
        <sharedTo>
            <role>CEO</role>
        </sharedTo>
    </sharingTerritoryRules>
</SharingRules>
```

The following is the definition of two account owner-based sharing rules in API version 32.0 and earlier. The file name corresponds to `Account.sharingRules` under the `accountSharingRules` directory. In this definition, ownerRules corresponds to AccountOwnerSharingRule.

```
<?xml version="1.0" encoding="UTF-8"?>
<AccountSharingRules xmlns="http://soap.sforce.com/2006/04/metadata">
    <ownerRules>
        <fullName>G1Dev_G2New</fullName>
        <sharedFrom>
            <group>G1Dev</group>
        </sharedFrom>
        <sharedTo>
            <group>G2New</group>
        </sharedTo>
        <accountAccessLevel>Read</caseAccessLevel>
```

```
        <caseAccessLevel>None</caseAccessLevel>
        <contactAccessLevel>Read</contactAccessLevel>
        <name>G1Dev_G2New</name>
        <opportunityAccessLevel>Edit</opportunityAccessLevel>
    </ownerRules>
    <ownerRules>
        <fullName>G2New_R1New</fullName>
        <sharedFrom>
            <group>G2New</group>
        </sharedFrom>
        <sharedTo>
            <roleAndSubordinates>R1New</roleAndSubordinates>
        </sharedTo>
        <accountAccessLevel>Edit</accountAccessLevel>
        <caseAccessLevel>Read</caseAccessLevel>
        <contactAccessLevel>Edit</contactAccessLevel>
        <name>G2New_R1New</name>
        <opportunityAccessLevel>None</opportunityAccessLevel>
    </ownerRules>
</AccountSharingRules>
```

The following is the definition of a user criteria-based sharing rule and a user membership-based sharing rule in API version 32.0 and earlier. The file name corresponds to `User.sharingRules` under the `userSharingRules` directory.

```
<?xml version="1.0" encoding="UTF-8"?>
<UserSharingRules xmlns="http://soap.sforce.com/2006/04/metadata">
    <criteriaBasedRules>
        <fullName>shareUsers2</fullName>
        <sharedTo>
            <group>Asia_Division</group>
        </sharedTo>
        <criteriaItems>
            <field>FirstName</field>
            <operation>equals</operation>
            <value>John</value>
        </criteriaItems>
        <name>shareUsers2</name>
        <userAccessLevel>Read</userAccessLevel>
    </criteriaBasedRules>
    <membershipRules>
        <fullName>shareUsers1</fullName>
        <sharedTo>
            <group>South_America_Division</group>
        </sharedTo>
        <sharedFrom>
            <group>Asia_Division</group>
        </sharedFrom>
        <name>shareUsers1</name>
        <userAccessLevel>Read</userAccessLevel>
    </membershipRules>
</UserSharingRules>
```

The following shows a sample `package.xml` file.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>ObjA__c.*</members>
        <name>SharingCriteriaRule</name>
    </types>
    <types>
        <members>ObjA__c.*</members>
        <name>SharingOwnerRule</name>
    </types>
    <version>64.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

1. BaseSharingRule

   This component is removed as of API version 33.0 and is available in earlier versions only. Use SharingBaseRule instead. Represents the base container for criteria-based and owner-based sharing rules.

2. CriteriaBasedSharingRule

   This component is removed as of API version 33.0 and is available in earlier versions only. Use SharingRules instead. Represents a criteria-based sharing rule. CriteriaBasedSharingRule enables you to share records based on specific criteria.

3. OwnerSharingRule

   Represents an ownership-based sharing rule. OwnerSharingRule enables you to share records owned by a set of users with another set, using rules that specify the access level of the target user group. This component is removed as of API version 33.0 and is available in earlier versions only.

BaseSharingRule

This component is removed as of API version 33.0 and is available in earlier versions only. Use SharingBaseRule instead. Represents the base container for criteria-based and owner-based sharing rules.

This type extends the Metadata metadata type and inherits its `fullName` field.

Note: You can't create a BaseSharingRule component directly. Use the components under the CriteriaBasedSharingRule or OwnerSharingRule metadata types instead.

Version

BaseSharingRule on page 2162 components are available in API version 24.0 and later.

Fields

Field	Field Type	Description
`sharedTo`	SharedTo	Required. Specifies who the record is shared with.
`fullName`	string	The unique identifier for API access.The `fullName` can contain only underscores and alphanumeric characters. It must be unique, begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. This field is inherited from the Metadata component.

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

CriteriaBasedSharingRule

This component is removed as of API version 33.0 and is available in earlier versions only. Use SharingRules instead. Represents a criteria-based sharing rule. CriteriaBasedSharingRule enables you to share records based on specific criteria.

It extends the BaseSharingRule metadata type and inherits its `sharedTo` field. For more information, see "Criteria-Based Sharing Rules Overview" in Salesforce Help.

Note: You can't create a CrteriaBasedSharingRule component directly. Use the child components instead.

Declarative Metadata File Suffix and Directory Location

CriteriaBasedSharingRule components are stored within the `SharingRules` component in the `criteriaBasedRules` field.

Version

CriteriaBasedSharingRule components are available in API version 24.0 and later.

Special Access Rules

As of Spring '20 and later, only users with the View Setup and Configuration permission can access this object, and only users with the Manage Sharing permission can edit this object.

Fields

The following information assumes that you're familiar with implementing sharing rules for standard objects and custom objects. For more information on these fields, see Sharing Settings in Salesforce Help.

Field	Field Type	Description
`criteriaItems`	FilterItem[]	List that represents the criteria for the sharing rule. The possible values are: • `field` • `operation` • `value`

AccountCriteriaBasedSharingRule

Represents a criteria-based sharing rule for accounts. It extends the CriteriaBasedSharingRule metadata type and inherits its `criteriaItems` field.

AccountCriteriaBasedSharingRule is used by the `criteriaBasedRules` field in AccountSharingRules.

Field	Field Type	Description
`accountAccessLevel`	`ShareAccessLevelNoNone` (enumeration of type string)	Required. A value that represents the level of access that the user or group has to the account. The possible values are: • `Read` • `Edit` • `All`
`booleanFilter`	string	Represents the filter logic of the sharing rule.
`caseAccessLevel`	`ShareAccessLevelNoAll` (enumeration of type string)	Required. A value that represents the level of access that the user or group has to cases associated with the account. The possible values are: • `None` • `Read` • `Edit`
`contactAccessLevel`	`ShareAccessLevelNoAll` (enumeration of type string)	Required. A value that represents the level of access that the user or group has to contacts associated with the account. The possible values are: • `None` • `Read` • `Edit`
`description`	string	Represents the description of the sharing rule. Maximum of 1000 characters. This field is available in API version 29.0 and later.
`name`	string	Required. Name for the sharing rule. Corresponds to **Label** in the user interface.

Field	Field Type	Description
`opportunityAccessLevel`	`ShareAccessLevelNoAll` (enumeration of type string)	Required. A value that represents the level of access that a target group is granted for any associated opportunity. The possible values are: • `None` • `Read` • `Edit`

CampaignCriteriaBasedSharingRule

Represents a criteria-based sharing rule for campaigns. It extends the CriteriaBasedSharingRule metadata type and inherits its `criteriaItems` field.

CampaignCriteriaBasedSharingRule is used by the `criteriaBasedRules` field in CampaignSharingRules.

Field	Field Type	Description
`booleanFilter`	string	Represents the filter logic of the sharing rule.
`description`	string	Represents the description of the sharing rule. Maximum of 1000 characters. This field is available in API version 29.0 and later.
`campaignAccessLevel`	`ShareAccessLevelNoNone` (enumeration of type string)	Required. A value that represents the level of access that a target group is granted for a campaign. The possible values are: • `Read` • `Edit` • `All`
`name`	string	Required. Name for the sharing rule. Corresponds to **Label** in the user interface.

CaseCriteriaBasedSharingRule

Represents a criteria-based sharing rule for cases. It extends the CriteriaBasedSharingRule metadata type and inherits its `criteriaItems` field.

CaseCriteriaBasedSharingRule is used by the `criteriaBasedRules` field in CaseSharingRules.

Field	Field Type	Description
`booleanFilter`	string	Represents the filter logic of the sharing rule.
`description`	string	Represents the description of the sharing rule. Maximum of 1000 characters. This field is available in API version 29.0 and later.

Field	Field Type	Description
`caseAccessLevel`	`ShareAccessLevelReadEdit` (enumeration of type string)	Required. A value that represents the level of access being granted for a case. The possible values are: • `Read` • `Edit`
`name`	string	Required. Name for the sharing rule. Corresponds to **Label** in the user interface.

ContactCriteriaBasedSharingRule

Represents a criteria-based sharing rule for contacts. It extends the CriteriaBasedSharingRule metadata type and inherits its `criteriaItems` field.

ContactCriteriaBasedSharingRule is used by the `criteriaBasedRules` field in ContactSharingRules.

Field	Field Type	Description
`booleanFilter`	string	Represents the filter logic of the sharing rule.
`description`	string	Represents the description of the sharing rule. Maximum of 1000 characters. This field is available in API version 29.0 and later.
`contactAccessLevel`	`ShareAccessLevelReadEdit` (enumeration of type string)	Required. A value that represents the level of access being granted to the target group, role, or user for a contact. The possible values are: • `Read` • `Edit`
`name`	string	Required. Name for the sharing rule. Corresponds to **Label** in the user interface.

LeadCriteriaBasedSharingRule

Represents a criteria-based sharing rule for leads. It extends the CriteriaBasedSharingRule metadata type and inherits its `criteriaItems` field.

LeadCriteriaBasedSharingRule is used by the `criteriaBasedRules` field in LeadSharingRules.

Field	Field Type	Description
`booleanFilter`	string	Represents the filter logic of the sharing rule.
`description`	string	Represents the description of the sharing rule. Maximum of 1000 characters. This field is available in API version 29.0 and later.

Field	Field Type	Description
`leadAccessLevel`	`ShareAccessLevelReadEdit` (enumeration of type string)	Required. A value that represents the level of allowed access. The possible values are: • `Read` • `Edit`
`name`	string	Required. Name for the sharing rule. Corresponds to **Label** in the user interface.

OpportunityCriteriaBasedSharingRule

Represents a criteria-based sharing rule for opportunities. It extends the CriteriaBasedSharingRule metadata type and inherits its `criteriaItems` field.

OpportunityCriteriaBasedSharingRule is used by the `criteriaBasedRules` field in OpportunitySharingRules.

Field	Field Type	Description
`booleanFilter`	string	Represents the filter logic of the sharing rule.
`description`	string	Represents the description of the sharing rule. Maximum of 1000 characters. This field is available in API version 29.0 and later.
`opportunityAccessLevel`	`ShareAccessLevelReadEdit` (enumeration of type string)	Required. A value that represents the level of allowed access. The possible values are: • `Read` • `Edit`
`name`	string	Required. Name for the sharing rule. Corresponds to **Label** in the user interface.

CustomObjectCriteriaBasedSharingRule

Represents a criteria-based sharing rule for custom objects. It extends the CriteriaBasedSharingRule metadata type and inherits its `criteriaItems` field.

CustomObjectCriteriaBasedSharingRule is used by the `criteriaBasedRules` field in CustomObjectSharingRules.

Field	Field Type	Description
`accessLevel`	string	Required. A value that represents the type of allowed sharing. The possible values are: • `Read` • `Edit` • `All`

Field	Field Type	Description
booleanFilter	string	Represents the filter logic of the sharing rule.
description	string	Represents the description of the sharing rule. Maximum of 1000 characters. This field is available in API version 29.0 and later.
name	string	Required. Name for the sharing rule. Corresponds to **Label** in the user interface.

UserCriteriaBasedSharingRule

Represents a criteria-based sharing rule for users. It extends the CriteriaBasedSharingRule metadata type and inherits its `criteriaItems` field.

UserCriteriaBasedSharingRule is used by the `criteriaBasedRules` field in UserSharingRules.

Field	Field Type	Description
booleanFilter	string	Represents the filter logic of the sharing rule.
description	string	Represents the description of the sharing rule. Maximum of 1000 characters. This field is available in API version 29.0 and later.
name	string	Required. Name for the sharing rule. Corresponds to **Label** in the user interface.
userAccessLevel	`ShareAccessLevelReadEdit` (enumeration of type string)	Required. A value that represents the type of allowed sharing. The possible values are: • `Read` • `Edit`

Declarative Metadata Sample Definition

The following is the definition of two owner-based sharing rules and one criteria-based sharing rule containing two criteria items. The file name corresponds to the Account.sharingRules file under the accountSharingRules directory.

```
<?xml version="1.0" encoding="UTF-8"?>
<AccountSharingRules xmlns="http://soap.sforce.com/2006/04/metadata">
  <ownerRules>
    <fullName>G1Dev_G2New</fullName>
    <sharedTo>
      <group>G2New</group>
    </sharedTo>
    <sharedFrom>
      <group>G1Dev</group>
    </sharedFrom>
    <accountAccessLevel>Read</accountAccessLevel>
```

```
    <caseAccessLevel>None</caseAccessLevel>
    <contactAccessLevel>Read</contactAccessLevel>
  </ownerRules>
    <fullName>G2New_R1New</fullName>
    <sharedTo>
      <roleAndSubordinates>R1New</roleAndSubordinates>
    </sharedTo>
    <sharedFrom>
      <group>G2New</group>
    </sharedFrom>
    <accountAccessLevel>Edit</accountAccessLevel>
    <caseAccessLevel>Read</caseAccessLevel>
    <contactAccessLevel>Edit</contactAccessLevel>
    <name>G2New_R1New</name>
    <opportunityAccessLevel>None</opportunityAccessLevel>
  </ownerRules>
  <criteriaBasedRules>
    <fullName>AccountCriteria</fullName>
    <sharedTo>
      <group>G1</group>
    </sharedTo>
    <criteriaItems>
      <field>BillingCity</field>
      <operation>equals</operation>
      <value>San Francisco</value>
    </criteriaItems>
    <criteriaItems>
      <field>MyChkBox__c</field>
      <operation>notEqual</operation>
      <value>False</value>
    </criteriaItems>
    <accountAccessLevel>Read</accountAccessLevel>
    <booleanFilter>1 OR 2</booleanFilter>
    <caseAccessLevel>None</caseAccessLevel>
    <contactAccessLevel>Read</contactAccessLevel>
    <name>AccountCriteria</name>
    <opportunityAccessLevel>None</opportunityAccessLevel>
  </criteriaBasedRules>
</AccountSharingRules>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

OwnerSharingRule

Represents an ownership-based sharing rule. OwnerSharingRule enables you to share records owned by a set of users with another set, using rules that specify the access level of the target user group. This component is removed as of API version 33.0 and is available in earlier versions only.

OwnerSharingRule extends the BaseSharingRule metadata type and inherits its SharedTo field. For more information, see "Sharing Rules" in the Salesforce online help.

Note: You can't create a OwnerSharingRule component directly. Use the child components instead.

Declarative Metadata File Suffix and Directory Location

OwnerSharingRules components are stored within the `SharingRules` component in the `ownerRules` field.

Version

OwnerSharingRules components are available in API version 24.0 and later.

Special Access Rules

As of Spring '20 and later, only users with the View Setup and Configuration permission can access this object, and only users with the Manage Sharing permission can edit this object.

Fields

The following information assumes that you are familiar with implementing sharing rules for standard objects and custom objects. For more information on these fields, see "Sharing Settings" in the Salesforce online help.

Field	Field Type	Description
`sharedFrom`	SharedTo	Required. Specifies the record owners.
`sharedTo`	SharedTo	Required. Specifies who the record should be shared with.
`fullName`	string	The unique identifier for API access. The `fullName` can contain only underscores and alphanumeric characters. It must be unique, begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. This field is inherited from the Metadata component.

AccountOwnerSharingRule

Represents a sharing rule for an account with users other than the owner. It extends the OwnerSharingRule metadata type and inherits its `fullName`, `sharedFrom`, and `sharedTo` fields.

AccountOwnerSharingRule is used by the `ownerRules` field in AccountSharingRules.

Field	Field Type	Description
`accountAccessLevel`	`ShareAccessLevelNoNone` (enumeration of type string)	Required. A value that represents the level of access that a group or role has to the account. The possible values are: • `Read` • `Edit` • `All`

Field	Field Type	Description
caseAccessLevel	ShareAccessLevelNoAll (enumeration of type string)	Required. A value that represents the level of access that a group or role has to cases associated with the account. The possible values are: • None • Read • Edit
contactAccessLevel	ShareAccessLevelNoAll (enumeration of type string)	Required. A value that represents the level of access that a group or role has to contacts associated with the account. The possible values are: • None • Read • Edit
description	string	Represents the description of the sharing rule. Maximum of 1000 characters.This field is available in API version 29.0 and later.
name	string	Required. Name for the sharing rule. Corresponds to **Label** in the user interface.
opportunityAccessLevel	ShareAccessLevelNoAll (enumeration of type string)	Required. A value that represents the level of access that a group or role is granted for any associated opportunity. The possible values are: • None • Read • Edit

CampaignOwnerSharingRule

Represents a sharing rule for a campaign with users other than the owner. It extends the OwnerSharingRule metadata type and inherits its fullName, sharedFrom, and sharedTo fields.

CampaignOwnerSharingRule is used by the ownerRules field in CampaignSharingRules.

Field	Field Type	Description
campaignAccessLevel	ShareAccessLevelNoNone (enumeration of type string)	A value that represents the level of access that a group or role is granted for a campaign. The possible values are: • Read • Edit • All
description	string	Represents the description of the sharing rule. Maximum of 1000 characters.This field is available in API version 29.0 and later.

Field	Field Type	Description
`name`	string	Name for the sharing rule. Corresponds to **Label** in the user interface.

CaseOwnerSharingRule

Represents a sharing rule for a case with users other than the owner. It extends the OwnerSharingRule metadata type and inherits its `fullName`, `sharedFrom`, and `sharedTo` fields.

CaseOwnerSharingRule is used by the `ownerRules` field in CaseSharingRules. All the following fields are required.

Field	Field Type	Description
`caseAccessLevel`	`ShareAccessLevelReadEdit` (enumeration of type string)	Required. A value that represents the level of access that a group or role is granted for a case. The possible values are: • `Read` • `Edit`
`description`	string	Represents the description of the sharing rule. Maximum of 1000 characters.This field is available in API version 29.0 and later.
`name`	string	Required. Name for the sharing rule. Corresponds to **Label** in the user interface.

ContactOwnerSharingRule

Represents a sharing rule for a contact with users other than the owner. It extends the OwnerSharingRule metadata type and inherits its `fullName`, `sharedFrom`, and `sharedTo` fields.

ContactOwnerSharingRule is used by the `ownerRules` field in ContactSharingRules.

Field	Field Type	Description
`contactAccessLevel`	`ShareAccessLevelReadEdit` (enumeration of type string)	Required. A value that represents the level of access that a group or role is granted for a contact. The possible values are: • `Read` • `Edit`
`description`	string	Represents the description of the sharing rule. Maximum of 1000 characters.This field is available in API version 29.0 and later.
`name`	string	Required. Name for the sharing rule. Corresponds to **Label** in the user interface.

LeadOwnerSharingRule

Represents a sharing rule for a lead with users other than the owner. It extends the OwnerSharingRule metadata type and inherits its fullName, `sharedFrom`, and `sharedTo` fields.

LeadOwnerSharingRule is used by the `ownerRules` field in LeadSharingRules.

Field	Field Type	Description
`leadAccessLevel`	`ShareAccessLevelReadEdit` (enumeration of type string)	Required. A value that represents the level of access that a group or role is granted for a lead. The possible values are: • `Read` • `Edit`
`description`	string	Represents the description of the sharing rule. Maximum of 1000 characters.This field is available in API version 29.0 and later.
`name`	string	Required. Required. Name for the sharing rule. Corresponds to **Label** in the user interface.

OpportunityOwnerSharingRule

Represents a sharing rule for an opportunity with users other than the owner. It extends the OwnerSharingRule metadata type and inherits its `fullName`, `sharedFrom`, and `sharedTo` fields.

OpportunityOwnerSharingRule is used by the `ownerRules` field in OpportunitySharingRules.

Field	Field Type	Description
`name`	string	Required. Name for the sharing rule. Corresponds to **Label** in the user interface.
`description`	string	Represents the description of the sharing rule. Maximum of 1000 characters.This field is available in API version 29.0 and later.
`opportunityAccessLevel`	`ShareAccessLevelReadEdit` (enumeration of type string)	Required. A value that represents the level of access that a group or role is granted for an opportunity. The possible values are: • `Read` • `Edit`

AccountTerritorySharingRule

Represents a rule for sharing an account within a territory. It extends the OwnerSharingRule metadata type and inherits its `fullName`, `sharedFrom`, and `sharedTo` fields.

AccountTerritorySharingRule is used by the `ownerRules` field in AccountTerritorySharingRules.

Field	Field Type	Description
`accountAccessLevel`	`ShareAccessLevelNoNone` (enumeration of type string)	Required. A value that represents the level of access that a Territory or TerritoryAndSubordinates group is granted for an account territory. The possible values are: • `Read` • `Edit`

Field	Field Type	Description
		• `All`
`caseAccessLevel`	`ShareAccessLevelNoAll` (enumeration of type string)	Required. A value that represents the level of access that a Territory or TerritoryAndSubordinates group is granted for all child cases to an account. The possible values are: • `None` • `Read` • `Edit`
`contactAccessLevel`	`ShareAccessLevelNoAll` (enumeration of type string)	Required. A value that represents the level of access that a Territory or TerritoryAndSubordinates group is granted for all related contacts on an account. The possible values are: • `None` • `Read` • `Edit`
`description`	string	Represents the description of the sharing rule. Maximum of 1000 characters.This field is available in API version 29.0 and later.
`name`	string	Required. Name for the sharing rule. Corresponds to **Label** in the user interface.
`opportunityAccessLevel`	`ShareAccessLevelNoAll` (enumeration of type string)	Required. A value that represents the level of access that a Territory or TerritoryAndSubordinates group is granted for all opportunities associated with an account. The possible values are: • `None` • `Read` • `Edit`

CustomObjectOwnerSharingRule

Represents a sharing rule for custom objects. It extends the OwnerSharingRule metadata type and inherits its `fullName`, `sharedFrom`, and `sharedTo` fields.

CustomObjectOwnerSharingRule is used by the `ownerRules` field in CustomObjectSharingRules.

Field	Field Type	Description
`accessLevel`	string	Required. A value that represents the level of access that a group or role is granted to a custom object. The possible values are: • `Read` • `Edit` • `All`

Field	Field Type	Description
description	string	Represents the description of the sharing rule. Maximum of 1000 characters. This field is available in API version 29.0 and later.
name	string	Required. Name for the sharing rule. Corresponds to **Label** in the user interface.

UserMembershipSharingRule

Represents a sharing rule to share members of a group with another group of users. It extends the OwnerSharingRule metadata type and inherits its `fullName`, `sharedFrom`, and `sharedTo` fields.

UserMembershipSharingRule is used by the `ownerRules` field in UserSharingRules on page 2159.

Field	Field Type	Description
description	string	Represents the description of the sharing rule. Maximum of 1000 characters.This field is available in API version 29.0 and later.
name	string	Required. Name for the sharing rule. Corresponds to **Label** in the user interface.
userAccessLevel	`ShareAccessLevelReadEdit` (enumeration of type string)	Required. A value that represents the level of access that a group or role is granted for a user. The possible values are: • `Read` • `Edit`

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SharingSet

Represents a sharing set. A sharing set defines an access mapping that grants portal or community users access to objects that are associated with their accounts or contacts.

This type extends the Metadata metadata type and inherits its `fullName` field.

For example, you can grant portal or community users access to all cases related to their account record. Similarly, you can grant portal or community users access to all cases related to a parent account that is identified on the user's account record.

File Suffix and Directory Location

SharingSet components have the suffix `.sharingSet` and are stored in the `sharingSets` folder.

Version

SharingSet components are available in API version 30.0 and later.

Special Access Rules

As of Spring '20 and later, only users with the View Setup and Configuration permission can access this object, and only users with the Manage Sharing permission can edit this object. To create or update sharing sets, you need the Customize Application permission.

Sharing sets are available with these licenses.

- Authenticated Website
- Customer Community Login
- Customer Community Plus
- Partner Community
- Customer Community User
- High Volume Customer Portal
- High Volume Portal
- Overage Authenticated Website User
- Overage High Volume Customer Portal User

Fields

Field Name	Field Type	Description
`accessMappings`	AccessMapping[]	A list of access mappings on a sharing set.
`description`	string	The sharing set description. Limit: 255 characters.
`name`	string	Required. The unique identifier for API access. Corresponds to **Sharing Set Name** on the user interface.
`profiles`	string[]	The profiles of users that are granted access to the target objects. Profiles must be associated with a license that can use sharing sets. See Special Access Rules for more information.

AccessMapping

AccessMapping represents an access mapping in the sharing set, which grants access to a target object by looking up to an account or contact associated with the user.

You can grant portal users access to a target object, or to both a target object and its associated objects, such as an account and its contacts and cases.

Field Name	Field Type	Description
`accessLevel`	string	Required. The target object access level granted to the portal user. Valid values are: • `Read`

Field Name	Field Type	Description
		• `Edit`
`objectField`	string	Required. A lookup to the target object, which supports standard or custom fields, or an ID. For accounts or cases associated with entitlements, use `Entitlement.Account` or `Entitlement.Case`.
`object`	string	Required. The target object to which the portal user is gaining access, and refers to one of the following: • `Account` • `Campaign` • `Contact` • `Case` • Custom Objects (for example, `ObjA__c`) • `Opportunity` • `Order` • `ServiceContract` • `User` • `WorkOrder` Portal users gain access to all order entitlements and order items under an account to which they have access.
`userField`	string	Required. The user's lookup to an account, contact, or a standard or custom field derived from an account or contact. Either the user or the user's manager can be used in the lookup. Valid values are: • `Account` • `Account.`***`Field`*** • `Contact` • `Contact.`***`Field`*** • `Contact.RelatedAccount` • `Manager.Account` • `Manager.Contact` ***`Field`*** refers to a standard or custom field based on an account or contact.

Declarative Metadata Sample Definition

The following is an example of a SharingSet component that grants users access to all contacts whose `ReportsTo` fields match the users' contacts.

```
<?xml version="1.0" encoding="UTF-8"?>
<SharingSet xmlns="http://soap.sforce.com/2006/04/metadata">
  <accessMappings>
    <accessLevel>Read</accessLevel>
```

```
    <objectField>ReportsTo</objectField>
    <object>Contact</object>
    <userField>Contact</userField>
  </accessMappings>
  <description>User Access Mapping</description>
  <name>User</name>
  <profiles>customer community user</profiles>
</SharingSet>
```

The following is an example of a SharingSet component that grants users access to all cases that are related to an entitlement, which is associated with the user's account.

```
<?xml version="1.0" encoding="UTF-8"?>
<SharingSet xmlns="http://soap.sforce.com/2006/04/metadata">
  <name>Case</name>
   <accessMappings>
    <accessLevel>Edit</accessLevel>
    <objectField>Entitlement.Account</objectField>
    <object>Case</object>
    <userField>Account</userField>
  </accessMappings>
</SharingSet>
```

The following is an example of a SharingSet component with a list of access mappings.

```
<?xml version="1.0" encoding="UTF-8"?>
<SharingSet xmlns="http://soap.sforce.com/2006/04/metadata">
  <description>This is a basic sharing set with several access mappings.</description>
  <name>Basic</name>
  <profiles>customer community user</profiles>
  <accessMappings>
    <accessLevel>Read</accessLevel>
    <objectField>Id</objectField>
    <object>Account</object>
    <userField>Account</userField>
  </accessMappings>
  <accessMappings>
    <accessLevel>Edit</accessLevel>
    <objectField>Account</objectField>
    <object>Contact</object>
    <userField>Account</userField>
  </accessMappings>
  <accessMappings>
    <accessLevel>Edit</accessLevel>
    <objectField>Contact</objectField>
    <object>Case</object>
    <userField>Contact</userField>
  </accessMappings>
  <accessMappings>
    <accessLevel>Read</accessLevel>
    <objectField>AccountLookup__c</objectField>
    <object>HVPUAccessible__c</object>
    <userField>Account</userField>
  </accessMappings>
</SharingSet>
```

The following is an example `package.xml` that references the previous definition.

```
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
  <fullName>SharingSetBasic</fullName>
  <types>
    <members>HVPUAccessible__c.AccountLookup__c</members>
    <members>HVPUAccessible__c.ContactLookup__c</members>
    <name>CustomField</name>
  </types>
  <types>
    <members>HVPUAccessible__c</members>
    <name>CustomObject</name>
  </types>
  <types>
    <members>Basic</members>
    <name>SharingSet</name>
  </types>
  <version>30.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SiteDotCom

Represents a site for deployment.

SiteDotCom extends the MetadataWithContent type and inherits its `fullName` and `content` fields.

Declarative Metadata File Suffix and Directory Location

SiteDotCom components are stored in the `siteDotComSites` directory of the corresponding package directory.

The file name for the metadata `.xml` file is `[sitename]1.site-meta.xml`. The file name for the site file is `[sitename]1.site`.

When a Lightning site is created, two sites are made behind the scenes: CustomSite (of type ChatterNetwork) and SiteDotComSite (of type ChatterNetworkPicasso). These sites are named, respectively, *<site_name>* and *<site_name>1*. The corresponding MDAPI file names are *<site_name>*`.site-meta.xml` and *<site_name>*`1.site-meta.xml`. 1 is appended to the SiteDotComSite type to keep the name unique from the corresponding CustomSite site.

Note: There is a file size limitation when using the Metadata API to deploy a site from sandbox to production. The assets in the `.site` file can't be larger than 40 MB. The site gets created, but the assets show in the new site as broken. To fix the assets, export the assets from the sandbox environment separately and then import them into your new site.

Version

`SiteDotCom` components are available in API version 30.0 and later.

Fields

Field	Field Type	Description
`label`	string	The name of the site that you're deploying.
`siteType`	(enumeration of type string)	Required. Identifies whether the site is a `ChatterNetworkPicasso` site for Experience Cloud Sites, or a `Siteforce` site for Site.com sites.

Declarative Metadata Sample Definition

Here are two examples of a SiteDotCom XML definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<SiteDotCom xmlns="http://soap.sforce.com/2006/04/metadata">
    <label>testsite</label>
    <siteType>Siteforce</siteType>
</SiteDotCom>


<?xml version="1.0" encoding="UTF-8"?>
<SiteDotCom xmlns="http://soap.sforce.com/2006/04/metadata">
    <label>testCommunity</label>
    <siteType>ChatterNetworkPicasso</siteType>
</SiteDotCom>
```

Usage

You can only deploy a `.site` file retrieved in Summer '19 or later. Older files aren't supported.

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

Skill

Represents the settings for a skill used for field service or to route chats to agents in Chat, such as the name of the skill and which agents the skills are assigned to.

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

Skill values are stored in the `<developer_name>.skill` file in the `skills` directory.

Version

Skill is available in API version 28.0 and later.

Fields

Field Name	Field Type	Description
`assignments`	SkillAssignments	Specifies how skills are assigned to Chat users. Skills can be assigned to sets of users or sets of profiles.
`description`	string	Specifies the description of the skill. This field is available in API version 38.0 and later.
`label`	string	Specifies the name of the skill.
`skillType`	string	Specifies the skill type, such as language or department, associated with the skill. This field is available in API version 58.0 and later.

SkillAssignments

Represents which users and user profiles to whom specific skills are assigned.

Fields

Field Name	Field Type	Description
`profiles`	SkillProfileAssignments	Specifies the profiles that are associated with a specific skill.
`users`	SkillUserAssignments	Specifies the users that are associated with a specific skill.

SkillProfileAssignments

Represents the profiles that are associated with a specific skill.

Fields

Field Name	Field Type	Description
`profile`	string	Specifies the custom name of the profile associated with a specific skill.

SkillUserAssignments

Represents the users that are associated with a specific skill.

Fields

Field Name	Field Type	Description
`user`	string	Specifies the username of the user associated with a specific skill.

Declarative Metadata Sample Definition

This is a sample of a `skill` file.

```
<?xml version="1.0" encoding="UTF-8"?>
<Skill xmlns="http://soap.sforce.com/2006/04/metadata">
    <label>My Skill 1</label>
    <assignments>
        <profiles>
            <profile>LiveAgentOperator</profile>
            <profile>LiveAgentSupervisor</profile>
        </profiles>
        <users>
            <user>jdoe@acme.com</user>
        </users>
    </assignments>
</Skill>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

StandardValueSet

Represents the set of values in a standard picklist field. This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

StandardValueSet components have the suffix `.standardValueSet` and are stored in the `standardValueSets` folder.

Version

StandardValueSet components are available in API version 38.0 and later.

Fields

Field Name	Field Type	Description
`groupingStringEnum`	string	Groups picklist and enumerated values. For example, for the picklist values of the `Status` field on the Service Appointment object, `Done` and `Finished` can both have a grouping string of `Completed`. Available in API version 41.0 and later.
`sorted`	boolean	Required. Indicates whether a global value set is sorted in alphabetical order. By default, this value is `false`.
`standardValue`	StandardValue[]	Defines each value in a standard picklist's value set. The `groupingString` value is available in API version 38.0 and later. When you deploy a StandardValueSet, this array must contain at least one picklist value. Otherwise, you receive an error. Note: When setting `standardValue` on Record Types, including person account record types, new picklist values loaded into your organization through the Metadata API don't display in the picklist UI by default. For users to see the new values, go to the Record Types list for the object containing the picklist field, click **Edit**, and add the new value to the Selected Fields list.

Declarative Metadata Sample Definition

The following example shows a StandardValueSet component that's defined as the Stage standard picklist on a customized opportunity object.

```
<?xml version="1.0" encoding="UTF-8"?>
<StandardValueSet xmlns="http://soap.sforce.com/2006/04/metadata">
    <fullName>OpportunityStage</fullName> <!-- Enum name -->
    <standardValue>
       <fullName>Closed Abandoned</fullName>
    </standardValue>
    <standardValue>
       <fullName>Closed Won</fullName>
    </standardValue>
    <standardValue>
       <fullName>Closed Lost</fullName>
    </standardValue>
</StandardValueSet>

<CustomObject>
    <fullName>Opportunity</fullName>
    <fields>
       <fullName>StageName</fullName> <!-- field name -->
       <label>Stage</label>
       <type>Picklist</type>
    </fields>
    <label>ObjectWithValueSet</label>
```

```
    <pluralLabel>ObjectWithValueSet</pluralLabel>
    <sharingModel>ReadWrite</sharingModel>
</CustomObject>
```

For a list of standard value set names for standard picklists, see StandardValueSet Names and Standard Picklist Fields.

Wildcard Support in the Manifest File

This metadata type doesn't support the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

StandardValueSetTranslation

Contains details for a standard picklist translation. It returns a translated standard value set.This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

StandardValueSetTranslation components have the suffix `.standardValueSetTranslation` and are stored in the `standardValueSetTranslations` folder.

Translations are stored in a file with a format of `ValueSetName-lang.standardValueSetTranslation`, where *`ValueSetName`* is the global value set's name, and lang is the translation language.

Version

StandardValueSetTranslation components are available in API version 38.0 and later.

Fields

Field	Field Type	Description
`valueTranslation`	ValueTranslation[]	A list of values from global value sets to be translated.

Declarative Metadata Sample Definition

The following is an example of a StandardValueSetTranslation component. When a value isn't translated, its translation becomes a comment that's paired with its label.

```
<?xml version="1.0" encoding="UTF-8"?>
<StandardValueSetTranslation xmlns="http://soap.sforce.com/2006/04/metadata">
    <valueTranslation>
        <masterLabel>Cold</masterLabel>
```

```
        <translation><!-- Cold --></translation>
    </valueTranslation>
    <valueTranslation>
        <masterLabel>Hot</masterLabel>
        <translation><!-- Hot --></translation>
    </valueTranslation>
    <valueTranslation>
        <masterLabel>Warm</masterLabel>
        <translation><!-- Warm --></translation>
    </valueTranslation>
</StandardValueSetTranslation>
```

The following is an example `package.xml` that references the StandardValueSetTranslation definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
        <types>
        <members>AccountRating-fr</members>
        <name>StandardValueSetTranslation</name>
    </types>
    <version>38.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SEE ALSO:

Translations

StaticResource

Represents a static resource file, often a code library in a ZIP file. Static resources allow you to upload content that you can reference in a Visualforce page, including archives (such as .zip and .jar files), images, style sheets, JavaScript, and other files. Static resources can be used only within your Salesforce org, so you can't host content here for other apps or websites.

This type extends the MetadataWithContent metadata type and inherits its `content` and `fullName` fields.

File Suffix and Directory Location

The file suffix is `.resource` for the template file. The accompanying metadata file is named *`resource`*`-meta.xml`.

Static resource components are stored in the `staticresources` folder in the corresponding package directory.

Version

Static resources are available in API version 12.0 and later.

Fields

This metadata type contains the following fields:

Field Name	Field Type	Description
`cacheControl`	StaticResourceCacheControl (enumeration of type string)	Required. Indicates whether the static resource is marked with a public caching tag so that a third-party delivery client can cache the content. This field is available in API version 14.0. The valid values are: • Private • Public
`content`	base64Binary	The static resource content. Base 64-encoded binary data. Before making an API call, client applications must encode the binary attachment data as base64. Upon receiving a response, client applications must decode the base64 data to binary. This conversion is handled for you by a SOAP client. This field is inherited from the MetadataWithContent component.
`contentType`	string	Required. The content type of the file, for example text/plain.
`description`	string	The description of the static resource.
`fullName`	string	The static resource name. The name can only contain characters, letters, and the underscore (_) character. The name must start with a letter, and can't end with an underscore or contain two consecutive underscore characters. Inherited from the Metadata component, this field isn't defined in the WSDL for this component. It must be specified when creating, updating, or deleting. See create() to see an example of this field specified for a call.

Declarative Metadata Sample Definition

```
<?xml version="1.0" encoding="UTF-8"?>
<StaticResource xmlns="http://soap.sforce.com/2006/04/metadata">
    <contentType>text/plain</contentType>
    <description>Test Resource</description>
</StaticResource>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SustainabilityUom

Represents the unit of measure (UOM) values for custom fuel types in an org. Track fuel consumption and emission results with the flexibility to add custom fuel types and UOM values.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

SustainabilityUom components have the suffix `.sustainabilityUom` and are stored in the `sustainabilityUoms` folder.

Version

SustainabilityUom components are available in API version 56.0 and later.

Special Access Rules

The Net Zero Cloud permission set license is required to access this object along with the user access for carbon accounting and org access for custom fuels and UOMs.

Fields

Field Name	Description
`description`	**Field Type** string **Description** The description of the unit of measure.
`isProductUom`	**Field Type** boolean **Description** Indicates whether the unit of measure is for a product that the company has procured in its supply chain operations (`true`) or not (`false`). The default value is `false`.
`isProtected`	**Field Type** boolean **Description** An auto-generated value that doesn't impact the behavior of the metadata type. The default value is `false`.
`isStationaryAssetUom`	**Field Type** boolean **Description** Indicates whether the unit of measure is used in the stationary asset calculations (`true`) or (`false`).

Field Name	Description
	The default value is `false`.
`isVehicleAssetUom`	**Field Type** boolean **Description** Indicates whether the unit of measure is used in the vehicle asset calculations (`true`) or (`false`). The default value is `false`.
`masterLabel`	**Field Type** string **Description** Required. The label assigned to this object.
`unitType`	**Field Type** UnitType (enumeration of type string) **Description** Required. The type of unit used for conversions or calculations. Values are: • `Energy` • `Other` • `Volume` • `Weight`

Declarative Metadata Sample Definition

The following is an example of a SustainabilityUom component.

```
<?xml version="1.0" encoding="UTF-8"?>
<SustainabilityUom xmlns="http://soap.sforce.com/2006/04/metadata">
   <description>Weight in Grams</description>
   <isProductUom>true</isProductUom>
   <isProtected>false</isProtected>
   <isStationaryAssetUom>false</isStationaryAssetUom>
   <isVehicleAssetUom>false</isVehicleAssetUom>
   <masterLabel>Grams</masterLabel>
   <unitType>Weight</unitType>
</SustainabilityUom>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
   <fullName>Pkg</fullName>
   <types>
    <members>Grams</members>
    <name>SustainabilityUom</name>
   </types>
   <version>64.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SustnUomConversion

Represents information about the unit of measure (UOM) conversion for the custom fuel types defined by a customer in an org.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

SustnUomConversion components have the suffix `sustnUomConversion` and are stored in the `sustnUomConversions` folder.

Version

SustnUomConversion components are available in API version 57.0 and later.

Special Access Rules

The Net Zero Cloud permission set license is required to access this object along with the user access for carbon accounting and org access for custom fuels and UOMs.

Fields

Field Name	Description
`conversionFactor`	**Field Type** double

Field Name	Description
	Description Required. The conversion factor that's used to convert values from one unit of measure to another for the fuel type.
`fuelType`	**Field Type** string **Description** The name of the fuel type. Possible values are: • `AutogasLPG` • `Biodiesel` • `Biomass` • `CityGas` • `CompressedNaturalGasCNG` • `Cooling` • `Diesel` • `Electricity` • `Ethanol` • `FuelOil` • `Gasoline` • `Heat` • `HeavyOil` • `ITElectricity` • `JetFuel` • `Kerosene` • `LightOil` • `LiquidNaturalGasLNG` • `MobileDiesel` • `NaturalGas` • `Propane` • `Refrigerant` • `Steam`
`isProtected`	**Field Type** boolean **Description** An auto-generated value that doesn't impact the behavior of the metadata type. The default value is `false`.

Field Name	Description
masterLabel	**Field Type** string **Description** A user-friendly name for SustnUomConversion, which is defined when the SustnUomConversion is created.
sourceUom	**Field Type** string **Description** Required. The source unit of measure for the fuel type. Possible values are: • 1000m3 • GJ • GWh • Kiloliters • Liters • MJ • MMBtu • MWh • Therms • Tonnes • UkGallons • UsGallons • ccf • kG • kWh • kcal • lbs • longTons • m3 • shortTons
targetUom	**Field Type** string **Description** Required. The target unit of measure for the fuel type.

Field Name	Description
uomsKey	**Field Type** string **Description** The key associated with a unit of measure for the fuel type.

Declarative Metadata Sample Definition

The following is an example of a SustnUomConversion component.

```
<?xml version="1.0" encoding="UTF-8"?>
<SustnUomConversion xmlns="http://soap.sforce.com/2006/04/metadata">
    <conversionFactor>0.9</conversionFactor>
    <fuelType>Diesel</fuelType>
    <isProtected>false</isProtected>
    <masterLabel>KG_Liters</masterLabel>
    <sourceUom>KG</sourceUom>
    <targetUom>Liters</targetUom>
    <uomsKey>uomsKey</uomsKey>
</SustnUomConversion>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <fullName>Pkg</fullName>
    <types>
        <members>US_UK_Gallons</members>
        <members>Therms_kWh</members>
        <members>KG_Liters</members>
        <name>SustnUomConversion</name>
    </types>
    <version>57.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SvcCatalogCategory

Represents the grouping of individual catalog items in Service Catalog.

File Suffix and Directory Location

`SvcCatalogCategory` components have the suffix `category` and are stored in the `svcCatalogCategories` folder.

Version

SvcCatalogCategory components are available in API version 53.0 and later.

Fields

Field Name	Description
`image`	**Field Type** string **Description** The developer name of a content document to be displayed in the Service Catalog for this category.
`isActive`	**Field Type** boolean **Description** Indicates if a catalog category is active.
`isProtected`	**Field Type** boolean **Description** An auto-generated value. This value currently has no impact.
`masterLabel`	**Field Type** string **Description** Required. The primary label for the catalog category record.
`parentCategory`	**Field Type** string **Description** If provided, the name of another SvcCatalogCategory that this category should appear under. The parent category in this field can't have its own parent category. Categories can't have more than one level of nesting.
`sortOrder`	**Field Type** int **Description** Displays a set order for catalog category entities.

Declarative Metadata Sample Definition

The following is an example of a SvcCatalogCategory component.

```
<?xml version="1.0" encoding="UTF-8"?>
<SvcCatalogCategory xmlns="http://soap.sforce.com/2006/04/metadata">
    <image>AdobeStock_287068722</image>
    <isActive>true</isActive>
    <isProtected>false</isProtected>
    <masterLabel>Workplace Services</masterLabel>
    <sortOrder>4</sortOrder>
</SvcCatalogCategory>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SvcCatalogFulfillmentFlow

Represents the flow associated with a specific catalog item in the Service Catalog.

File Suffix and Directory Location

`SvcCatalogFulfillmentFlow` components have the suffix `fulfillmentFlow` and are stored in the `svcCatalogFulfillmentFlows` folder.

Version

SvcCatalogFulfillmentFlows components are available in API version 53.0 and later.

Fields

Field Name	Description
`description`	**Field Type** string **Description** Required. Free-text description of the fulfillment flow.
`flow`	**Field Type** string **Description** Required. The name of the flow represented by this SvcCatalogFulfillmentFlow.

Field Name	Description
`icon`	**Field Type** string **Description** Represents the details of an icon.
`isProtected`	**Field Type** boolean **Description** An auto-generated value. This value currently has no impact.
`items`	**Field Type** SvcCatalogFulfillFlowItem on page 2195[] **Description** The list of variables in the flow that can accept a value as input.
`masterLabel`	**Field Type** string **Description** Required. The primary label for the fulfillment flow record.

SvcCatalogFulfillFlowItem

Represents a variable in a fulfillment flow that can accept input. Describes what type of value it accepts.

Field Name	Description
`catalogInputVariable`	**Field Type** string **Description** Required. The FlowVariable the fulfillment flow property represents.
`displayType`	**Field Type** PropertyDisplayType (enumeration of type string) **Description** The display options available. Values are: • `Checkbox` • `Date` (available in API version 59.0 and later) • `DateTime` (available in API version 59.0 and later)

Field Name	Description
	• `Lookup` • `Number` • `Picklist` • `Queue` (available in API version 57.0 and later) • `Text`
`fieldDefinition`	**Field Type** string **Description** The name of a field in the object provided in `objectLookupDomain` that specifies the value for this variable. If `displayType` is `Picklist`, this value must be the name of a picklist field. If `displayType` is `Lookup` and `fieldLookupDomain` is `FieldDefinition`, this value must be the name of a relationship field.
`fieldLookupDomain`	**Field Type** string **Description** The name of a standard or custom object that specifies the domain of that lookup or picklist. This value is relevant only if `displayType` is `Lookup` or `Picklist`.
`isAdditionalQuestionsInputVariable`	**Field Type** boolean **Description** Determines if this variable accepts input for all additional questions that were asked to a user. This value can only be `true` if the `displayType` for this item is `Text`. Only one item per SvcCatalogFulfillmentFlow component can set this attribute to `true`.
`isRequired`	**Field Type** boolean **Description** Determines if the field is required for the related fulfillment flow to be executed.
`lookupDomainFieldType`	**Field Type** string **Description** This value specifies the fields for the object specified by `objectLookupDomain` that are displayed in the Catalog Builder by type. This value is only relevant if `displayType` is `Lookup` and `fieldLookupDomain` is `FieldDefinition`.
`masterLabel`	**Field Type** string

Field Name	Description
	Description Required. The primary label for the fulfillment flow record.
objectLookupDomain	**Field Type** string **Description** The name of a custom or standard object. If displayType is Lookup or Picklist, this value filters the available options to a specific object.

Declarative Metadata Sample Definition

The following is an example of a SvcCatalogFulfillmentFlow component.

```
<?xml version="1.0" encoding="UTF-8"?>
<SvcCatalogFulfillmentFlow xmlns="http://soap.sforce.com/2006/04/metadata">
    <description>Creates a Case record related to the Contact belonging to the current
User. If this will be used by Users without related Contacts, provide an Account Id below.
 This Account Id will be used instead of a Contact.</description>
    <flow>Create_Case_by_Record_Type</flow>
    <isProtected>false</isProtected>
    <items>
        <catalogInputVariable>Input_RecordTypeApiName</catalogInputVariable>
        <displayType>Text</displayType>
        <isAdditionalQuestionsInputVariable>false</isAdditionalQuestionsInputVariable>
        <isRequired>true</isRequired>
        <masterLabel>Record Type Developer Name</masterLabel>
    </items>
    <items>
        <catalogInputVariable>Input_AccountId</catalogInputVariable>
        <displayType>Lookup</displayType>
        <fieldDefinition>AccountId</fieldDefinition>
        <fieldLookupDomain>Account</fieldLookupDomain>
        <isAdditionalQuestionsInputVariable>false</isAdditionalQuestionsInputVariable>
        <isRequired>false</isRequired>
        <masterLabel>(Optional) Related Account</masterLabel>
        <objectLookupDomain>Contact</objectLookupDomain>
    </items>
    <items>
        <catalogInputVariable>Input_Origin</catalogInputVariable>
        <displayType>Picklist</displayType>
        <fieldDefinition>Origin</fieldDefinition>
        <isAdditionalQuestionsInputVariable>false</isAdditionalQuestionsInputVariable>
        <isRequired>true</isRequired>
        <masterLabel>Case Origin</masterLabel>
        <objectLookupDomain>Case</objectLookupDomain>
    </items>
```

```
    <items>
        <catalogInputVariable>Input_Priority</catalogInputVariable>
        <displayType>Picklist</displayType>
        <fieldDefinition>Priority</fieldDefinition>
        <isAdditionalQuestionsInputVariable>false</isAdditionalQuestionsInputVariable>
        <isRequired>false</isRequired>
        <masterLabel>Case Priority</masterLabel>
        <objectLookupDomain>Case</objectLookupDomain>
    </items>
    <items>
        <catalogInputVariable>Input_Status</catalogInputVariable>
        <displayType>Picklist</displayType>
        <fieldDefinition>Status</fieldDefinition>
        <isAdditionalQuestionsInputVariable>false</isAdditionalQuestionsInputVariable>
        <isRequired>true</isRequired>
        <masterLabel>Case Status</masterLabel>
        <objectLookupDomain>Case</objectLookupDomain>
    </items>
    <items>
        <catalogInputVariable>Input_Subject</catalogInputVariable>
        <displayType>Text</displayType>
        <isAdditionalQuestionsInputVariable>false</isAdditionalQuestionsInputVariable>
        <isRequired>true</isRequired>
        <masterLabel>Case Subject</masterLabel>
    </items>
    <items>
        <catalogInputVariable>Input_Description</catalogInputVariable>
        <displayType>Text</displayType>
        <isAdditionalQuestionsInputVariable>true</isAdditionalQuestionsInputVariable>
        <isRequired>false</isRequired>
        <masterLabel>Case Description</masterLabel>
    </items>
    <masterLabel>Create Case by Record Type</masterLabel>
</SvcCatalogFulfillmentFlow>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SvcCatalogItemDef

Represents the entity associated with a specific, individual service available in the Service Catalog.

File Suffix and Directory Location

`SvcCatalogItemDef` components have the suffix `catalogItem` and are stored in the `svcCatalogItems` folder.

Version

SvcCatalogItemDef components are available in API version 53.0 and later.

Fields

Field Name	Description
`apiVersion`	**Field Type** double **Description** The API version in which this catalog item was created. The value for this field updates based on the value of `fulfillmentFlow`. For catalog items created before version 57.0, the value for this field is `null`. Available in version 57.0 and later.
`catalogFilterCriteria`	**Field Type** SvcCatalogItemDefFiltrCrit[] **Description** The eligibility rule associated with a catalog item. Eligibility rules customize access to catalog items for different audiences, based on the User object. Available in API version 59.0 and later.
`categories`	**Field Type** SvcCatalogCategoryItem[] **Description** A list of catalog categories that contain this catalog item.
`dataCategories`	**Field Type** SvcCatalogItemDefDataCategorySelection[] **Description** A list of data categories for this catalog item. Available in API version 59.0 and later.
`description`	**Field Type** string **Description** Description of the catalog item.
`flow`	**Field Type** string **Description** The screen flow associated with the catalog item. Available in API version 53.0 to 58.0.
`fulfillmentFlow`	**Field Type** string

Field Name	Description
	Description Name of the related `SvcCatalogFulfillmentFlow` on page 2194, which represents the flow associated with a specific catalog item in the Service Catalog. Available in API version 56.0 and later.
`image`	**Field Type** string **Description** The developer name of a content document to be displayed in the Service Catalog for this item.
`inputs`	**Field Type** SvcCatalogItemAttribute[] **Description** Represents attributes of a catalog item version. Available in API version 57.0 and later.
`internalNotes`	**Field Type** string **Description** Intended to describe what the catalog item does and its implementation. That value is meant for other catalog builders.
`isAvailableToAllCustomers`	**Field Type** boolean **Description** Required. Controls catalog item access for internal users. To share with all internal users, set the value to `True`. This value corresponds to the **Allow Access for All Users** option for Internal Access in the Catalog Item Builder. Available in API version 61.0 and later.
`isFeatured`	**Field Type** boolean **Description** Determines if the catalog item is part of the featured catalog items.
`isGuestAccessible`	**Field Type** boolean **Description** Required. Controls catalog item access for guest users. To share with guests, set the value to `True`. This value corresponds to **Guest Visibility** option for External Access in the Catalog Item Builder. Available in API version 61.0 and later.

Field Name	Description
`isProtected`	**Field Type** boolean **Description** An auto-generated value. This value has no impact.
`masterLabel`	**Field Type** string **Description** Required. The primary label for the catalog item record.
`sharedTo`	**Field Type** SharedTo on page 2150 **Description** Describes how the catalog item is shared across multiple catalog categories. SvcCatalogItemDef only supports sharing with groups.
`status`	**Field Type** PublishStatusType (enumeration of type string) **Description** Required. Displays the publishing status of a catalog item. Values are: • `Deprecated` • `Draft` • `PendingChanges` • `Published`

SvcCatalogItemDefFiltrCrit

Represents the association of an eligibility rule with a catalog item. Eligibility rules customize access to catalog items for different audiences, based on the User object. Available in version 59.0 and later.

Field Name	Description
`svcCatalogFilterCriteria`	**Field Type** string **Description** The name of the associated `catalogFilerCriteria` eligibility filter.

SvcCatalogCategoryItem

Represents the assignment of this service to a category within the Service Catalog.

Field Name	Description
`isPrimaryCategory`	**Field Type** boolean **Description** Determines if the catalog category (`svcCatalogCategory`) is the primary category for this catalog item. Exactly one category per SvcCatalogItemDef component must set this attribute to true.
`sortOrder`	**Field Type** int **Description** The position of the catalog item relative to other catalog items in the catalog category.
`svcCatalogCategory`	**Field Type** string **Description** Required. The catalog category the catalog item is assigned to.

SvcCatalogItemDefDataCategorySelection

Represents a list of data categories for this catalog item. This subtype is available in API version 59.0 and later.

Field Name	Description
`category`	**Field Type** string **Description** API name of a data category.
`categoryGroup`	**Field Type** string **Description** API Name of a data category group.

SvcCatalogItemAttribute

Represents an attribute of a catalog item version. It can be a static input filled by the catalog builder user or additional questions that end users answer at runtime. Available in API version 57.0 and later.

Field Name	Description
`field`	**Field Type** string **Description** Applicable when the display type is Lookup/Reference.
`inputType`	**Field Type** SvcCatalogItemAttrDataType (enumeration of type string) **Description** Required. Values are: • `Attachment` • `Checkbox` • `Currency` • `Date` • `Datetime` • `DisplayText` • `Email` • `IPAddress` • `Integer` • `ListOfBoolean` • `ListOfDouble` • `ListOfInteger` • `ListOfMaps` • `ListOfString` • `Lookup` • `Map` • `MultilineText` • `Number` • `NumericScale` • `Percentage` • `Picklist` • `Queue` • `SingleCheckbox` (available in API version 59.0 and later) • `SinglelineText` • `Text` • `Toggle` (available in API version 59.0 and later) • `URL`

Field Name	Description
`inputVariable`	**Field Type** string **Description** References the input variable to which the attribute value is forwarded.
`isRequired`	**Field Type** boolean **Description** Determines if an answer is required for this question.
`label`	**Field Type** string **Description** A translatable label for rendering the attribute to users.
`maxValue`	**Field Type** double **Description** Applicable when the display type is slider.
`minValue`	**Field Type** double **Description** Applicable when the display type is slider.
`name`	**Field Type** string **Description** Required. Applicable when the display type is Lookup/Reference.
`object`	**Field Type** string **Description** A picklist object's custom API Name. Applies when `inputType` is set to `Picklist`.
`options`	**Field Type** SvcCatalogItemAttrDetail **Description** The values attached to an attribute of an item version.

Field Name	Description
`type`	**Field Type** SvcCatalogItemAttrType (enumeration of type string) **Description** Required. Type of the attribute; used to determine if it's a pre-filled input or questions to ask users. Values are: • `FulfillmentInput` • `UserQuestion`
`value`	**Field Type** string **Description** Attribute value defined by the catalog builder.

SvcCatalogItemAttrDetail

Represents the details for an attribute of an item version. Used for options displayed in picklist or checkbox groups.

Field Name	Description
`isDefault`	**Field Type** boolean **Description** Required. Marks the attribute detail as the default. Applicable when the input display type is picklist or checkbox.
`label`	**Field Type** string **Description** Required. Picklist option label when the input type is picklist or checkbox.
`value`	**Field Type** string **Description** Attribute value defined by the catalog builder.

Declarative Metadata Sample Definition

The following is an example of a SvcCatalogItemDef component.

```
<SvcCatalogItemDef xmlns="http://soap.sforce.com/2006/04/metadata">
    <apiVersion>57.0</apiVersion>
    <categories>
        <isPrimaryCategory>true</isPrimaryCategory>
        <sortOrder>3</sortOrder>
        <svcCatalogCategory>Category1</svcCatalogCategory>
    </categories>
    <dataCategories>
        <category>France</category>
        <categoryGroup>World</categoryGroup>
    </dataCategories>
    <masterLabel>Item Draft Update</masterLabel>
    <description>Item with a Draft state</description>
    <fulfillmentFlow>TestQuestions</fulfillmentFlow>
    <isFeatured>false</isFeatured>
    <isProtected>false</isProtected>
    <status>Published</status>
    <inputs>
        <name>Input1</name>
        <type>FulfillmentInput</type>
        <inputVariable>input1</inputVariable>
        <label>Input Static</label>
        <inputType>Text</inputType>
        <isRequired>false</isRequired>
    </inputs>
    <inputs>
        <type>UserQuestion</type>
        <inputType>Picklist</inputType>
        <isRequired>false</isRequired>
        <label>My First Question</label>
        <name>first_question</name>
        <options>
            <label>Option 1</label>
            <value>option_1</value>
            <isDefault>true</isDefault>
        </options>
        <options>
            <label>Option 2</label>
            <value>option_2</value>
            <isDefault>false</isDefault>
        </options>
        <options>
            <label>Option 3</label>
            <value>option_3</value>
            <isDefault>false</isDefault>
        </options>
    </inputs>
</SvcCatalogItemDef>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SynonymDictionary

Represents a set of synonym groups, which are groups of words or phrases that are treated as equivalent in users' searches. You can define synonym groups to optimize search results for acronyms, variations of product names, and other terminology unique to your organization.

Synonyms are available in Service Cloud features such as Salesforce Knowledge. This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

SynonymDictionary components have the suffix `.synonymDictionary` and are stored in the `synonymDictionaries` folder.

Version

SynonymDictionary components are available in API version 29.0 and later.

Special Access Rules

Synonyms must be enabled in your organization. Only users with the "Manage Synonyms" permission can access this object.

Fields

Field Name	Field Type	Description
`groups`	SynonymGroup	The synonym groups defined in this dictionary.
`isProtected`	boolean	Indicates whether this component is protected (`true`) or not (`false`). Protected components cannot be linked to or referenced by components created in the installing organization.
`label`	string	Required. Specifies the display name of the synonym dictionary.

SynonymGroup

Represents a group of synonymous words or phrases.

Field Name	Field Type	Description
`languages`	Language on page 2233	Required. Specifies the languages the synonym group applies to. If synonyms are specific to a single language, specify only that language. If the synonyms apply to multiple languages, specify multiple languages for one synonym group.

Field Name	Field Type	Description
`terms`	string	Required. A word or phrase synonymous with other terms in the group. Maximum of 50 characters. Minimum of two `terms` per group. Synonym groups are symmetric, which means that if oranges and apples are defined in a synonym group, a search for *`oranges`* will return a match for *`apples`*, and vice versa for a search for *`apples`*.

Declarative Metadata Sample Definition

The following is an example of a SynonymDictionary component:

```
<?xml version="1.0" encoding="UTF-8"?>
<SynonymDictionary xmlns="http://soap.sforce.com/2006/04/metadata">
    <groups>
        <languages>en_US</languages>
        <terms>Salesforce</terms>
        <terms>salesforce.com</terms>
        <terms>The Customer Company</terms>
        <terms>SFDC</terms>
    </groups>
    <groups>
        <languages>fr</languages>
        <terms>renault</terms>
        <terms>clio</terms>
    </groups>
    <label>Sample Dictionary</label>
</SynonymDictionary>
```

The following is an example `package.xml` that references the SynonymDictionary component.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>Sample Dictionary</members>
        <name>SynonymDictionary</name>
    </types>
    <version>64.0</version>
</Package>
```

Usage

If you have existing synonym groups defined before API version 29.0, your existing groups are associated with a default dictionary called `_Default`.

If you have a set of synonyms that require frequent updates, we recommend assigning the synonym group or groups to a dedicated dictionary with a small number of groups. Each time you deploy an existing dictionary, all of its synonym groups are overwritten. We don't support deploying updates to only a single synonym group within a dictionary.

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

Territory

Represents a territory.

Declarative Metadata File Suffix and Directory Location

The file suffix for territory components is `.territory` and components are stored in the `territories` directory of the corresponding package directory.

Version

Territory components are available in API version 24.0 and later.

Fields

This metadata type extends to subtype RoleOrTerritory.

Field Name	Field Type	Description
`accountAccessLevel`	string	Specifies whether users in this territory can access accounts that are assigned to this territory and are otherwise inaccessible. Valid values are: • `Read` • `Edit` • `All` If your organization's sharing model for accounts is Public Read/Write, valid values are only `Edit` and `All`. If no value is set for this field, this field value uses the default access level that is specified in the Manage Territory page in Setup. This field is available in API version 31.0 and later.
`fullName`	string	The unique identifier for API access. The `fullName` can contain only underscores and alphanumeric characters. It must be unique, begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. This field is inherited from the Metadata component. Corresponds to **Territory Name** in the user interface.
`parentTerritory`	string	The territory above this territory in the territory hierarchy.

Declarative Metadata Sample Definition

The following is the definition of a territory.

```
<?xml version="1.0" encoding="UTF-8"?>
<Territory xmlns="http://soap.sforce.com/2006/04/metadata">
    <accountAccessLevel>Edit</accountAccessLevel>
    <caseAccessLevel>Edit</caseAccessLevel>
    <contactAccessLevel>Edit</contactAccessLevel>
    <description>Sample Territory</description>
    <mayForecastManagerShare>false</mayForecastManagerShare>
    <name>T22name</name>
    <opportunityAccessLevel>Read</opportunityAccessLevel>
</Territory>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

Territory2

Represents the metadata associated with a sales territory. This type extends the Metadata metadata type and inherits its `fullName` field. Available if Sales Territories has been enabled.

File Suffix and Directory Location

Territory2 components have the suffix `territory2` and are stored in the `territories` folder under the folder for the corresponding Territory2Model.

Version

Territory2 components are available in API version 32.0 and later.

Special Access Rules

The Territory2Model object has a `State` field in the SOAP API. States include `Planning`, `Active`, `Archived`, and several other states, such as `Cloning`, that indicate that a process is underway. Users who do not have the Manage Territories permission can access territories that belong to the model in `Active` state. The Manage Territories permission is required for `deploy()` calls for all territory management entities. Using `retrieve()` without the Manage Territories permission returns only entities that belong to a Territory2Model in `Active` state. We recommend against retrieving without the Manage Territories permission because the call retrieves only partial data.

Fields

Field Name	Field Type	Description
`accountAccessLevel`	string	Specifies whether users in this territory can access accounts that are assigned to this territory and are otherwise inaccessible. Valid values are: • `Read` • `Edit` • `All` If your organization's sharing model for accounts is Public Read/Write, valid values are only `Edit` and `All`. If no value is set for this field, this field value uses the default access level that is specified in Territory2Settings as permitted by the organization's sharing settings.
`caseAccessLevel`	string	Specifies whether users in this territory can access cases that are assigned to this territory and are otherwise inaccessible. Valid values are: • `None` • `Read` • `Edit` Specify no value if your organization's sharing model for cases/opportunities is Public Read/Write. If no value is set for this field, this field value uses the default access level that is specified in Territory2Settings as permitted by the organization's sharing settings.
`contactAccessLevel`	string	Specifies whether users in this territory can access contacts that are assigned to this territory and are otherwise inaccessible. Valid values are: • `None` • `Read` • `Edit` Specify no value if your organization's sharing model for contacts is Public Read/Write or Controlled By Parent.
`customFields`	FieldValue	Values for custom fields defined on the Territory2 object and used by this territory. Their metadata is captured separately in CustomObject on page 663. Note the following: • Territory2 and Territory2Model objects do not handle values for Text Area (Long), Text Area (Rich), and text-encrypted custom fields. • Fields are referenced using their API names. Compound field types like Location appear as their constituent column fields. For example, `nnn_Latitude__s`, `nnn_Longitude__s` where "nnn" is the field name and the suffixes are the geolocation components. • Values of required custom fields are enforced during the `deploy()` operation.

Field Name	Field Type	Description
description	string	A description of the territory.
name	string	Required. The user interface label for the territory.
objectAccessLevels	Territory2AccessLevel[]	Represents the user access levels of an object associated to a territory. Available in API version 57.0 and later.
opportunityAccessLevel	string	Specifies whether users in this territory can access opportunities that are assigned to this territory and are otherwise inaccessible. Valid values are: • None • Read • Edit Specify no value if your organization's sharing model for cases/opportunities is Public Read/Write. If no value is set for this field, this field value uses the default access level that is specified in Territory2Settings as permitted by the organization's sharing settings.
parentTerritory	string	The name of the territory's parent. When you specify the parent territory, use the developer name. Do not use the "fully qualified" name. Custom fields with no values are retrieved with values of type: <value xsi:nil="true"/>. You can also use <value xsi:nil="true"/> syntax to remove existing values in custom fields.
ruleAssociations	Territory2RuleAssociation[]	Represents an object assignment rule and its association to a territory. Use the developer name of the rule.
territory2Type	string	Required. The territory type that the territory belongs to.

FieldValue

Represents the values of custom fields on the Territory2 object. Available in API version 32.0 and later.

Field Name	Field Type	Description
name	string	Required. The user interface label for the territory.
value	any type	The value of the field, which can also be null. The field type is specified in the XML and depends on the field value.

Territory2AccessLevel

Represents the association of an object access level to a territory. Available in API version 57.0 and later.

Field Name	Field Type	Description
accessLevel	string	Required. Valid values are: • Read • Edit • Transfer • All If your organization's sharing model for accounts is Public Read/Write, valid values are only Edit and All. If no value is set for this field, this field value uses the default access level that is specified in Territory2Settings as permitted by the organization's sharing settings.
objectType	string	Required. The type of object associated to the territory. For example, Lead.

Territory2RuleAssociation

Represents the association of an object assignment rule to a territory. Available in API version 32.0 and later.

Field Name	Field Type	Description
inherited	boolean	Required. Indicates whether the rule is inherited from a parent territory (true) or local to the current territory (false). Rule inheritance flows from the parent territory where the rule is created to the rule's descendent territories, if any, in the territory model hierarchy. A local rule is created within a single territory and affects that territory only.
ruleName	string	Required. The name of a rule associated with the territory. It isn't necessary to fully qualify ruleName because Metadata API assumes that the rule belongs to the same model as the territory.

Declarative Metadata Sample Definition

The following example shows the definition of a Territory2 component.

```
<?xml version="1.0" encoding="UTF-8"?>
<Territory2 xmlns="http://soap.sforce.com/2006/04/metadata"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <name>USA</name>
    <description>United States sales</description>
    <accountAccessLevel>Edit</accountAccessLevel>
    <opportunityAccessLevel>Read</opportunityAccessLevel>
    <caseAccessLevel>Edit</caseAccessLevel>
    <contactAccessLevel>Edit</contactAccessLevel>
    <parentTerritory>Worldwide_Sales</parentTerritory>
    <territory2Type>Geo</territory2Type>
    <objectAccessLevels>
        <accessLevel>All</accessLevel>
```

```
        <objectType>Lead</objectType>
    </objectAccessLevels>
    <ruleAssociations>
        <ruleName>AccRule1</name>
        <inherited>True</inherited>
     </ruleAssociations>
     <ruleAssociations>
        <ruleName>AccRule2</name>
        <inherited>False</inherited>
     </ruleAssociations>
     <customFields>
        <name>Activation_DateTime__c</name>
        <value xsi:type="xsd:dateTime">2014-07-16T05:05:00.000Z</value>
     </customFields>
     <customFields>
        <name>AutoNumber__c</name>
        <value xsi:type="xsd:string">T# 000001</value>
     </customFields>
     <customFields>
        <name>DeactivationDate__c</name>
        <value xsi:type="xsd:date">2016-07-12</value>
     </customFields>
     <customFields>
        <name>External_Id__c</name>
        <value xsi:type="xsd:string">AB2345</value>
     </customFields>
     <customFields>
        <name>ManagersPhone__c</name>
        <value xsi:nil="true"/>
     </customFields>
</Territory2>
```

The following is a `package.xml` sample. *`FY13`* and *`FY14`* represent the names of territory models and demonstrate that rules can have identical developer names within different models. A wildcard character (*) in place of the model name can be used to retrieve all rules in all models in an organization.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>FY13</members>
        <members>FY14</members>
        <name>Territory2Model</name>
    </types>

    <types>
        <members>FY13.USA</members>
        <members>FY13.Worldwide_Sales</members>
        <members>FY14.APAC</members>
        <members>FY14.USA</members>
        <name>Territory2</name>
    </types>

    <version>64.0</version>
</Package>
```

Usage

- Triggers defined on Territory2 do *not* fire during a `deploy()` operation unless there is a deployment failure. For example, when a child territory references a parent and deploys before the parent territory, the failed components try to deploy again one at a time, allowing triggers to run.
- Sales Territories components don't support packaging or change sets and aren't supported in CRUD calls.
- For unlocked packaging, `Territory2` requires packages without a namespace.

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

Territory2Model

Represents the metadata associated with a territory model in Sales Territories. This type extends the Metadata metadata type and inherits its `fullName` field.Available if Sales Territories has been enabled.

File Suffix and Directory Location

Territory2Model components have the suffix `territory2Model` and are stored in the `territory2Models` folder.

Version

Territory2Model components are available in API version 32.0 and later.

Special Access Rules

The Territory2Model object has a `State` field in the SOAP API. States include `Planning`, `Active`, `Archived`, and several other states, such as `Cloning`, that indicate that a process is underway. Users who do not have the Manage Territories permission can access models in `Active` state. The Manage Territories permission is required for `deploy()` calls for all territory management entities. Using `retrieve()` without the Manage Territories permission returns only entities that belong to a Territory2Model in `Active` state. We recommend against retrieving without the Manage Territories permission because the call retrieves only partial data.

Fields

Field Name	Field Type	Description
`customFields`	FieldValue	Custom fields defined on the Territory2Model object and used by this model. Their metadata is captured separately. • Territory2 and Territory2Model objects do not handle values for Text Area (Long), Text Area (Rich), and text-encrypted custom fields. • Fields are referenced using their API names. Compound field types like Location appear as their constituent column fields. For example,

Field Name	Field Type	Description
		`nnn_Latitude__s`, `nnn_Longitude__s` where "nnn" is the field name and the suffixes are the geolocation components. • Values of required custom fields are enforced during the `deploy()` operation.
`description`	string	A description for the territory model.
`name`	string	Required. The user interface label for the territory model.

Declarative Metadata Sample Definition

The following example shows the definition of a Territory2Model component.

```
<?xml version="1.0" encoding="UTF-8"?>
    <Territory2Model xmlns="http://soap.sforce.com/2006/04/metadata"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <name>FY13</name>
    <description>Geographic allocation</description>
    <customFields>
        <name>Activation_DateTime__c</name>
        <value xsi:type="xsd:dateTime">2014-07-16T05:05:00.000Z</value>
    </customFields>
    <customFields>
        <name>AutoNumber__c</name>
        <value xsi:type="xsd:string">M# 000001</value>
    </customFields>
    <customFields>
        <name>DeactivationDate__c</name>
        <value xsi:type="xsd:date">2016-07-12</value>
    </customFields>
    <customFields>
        <name>External_Id__c</name>
        <value xsi:nil="true"/>
    </customFields>
</Territory2Model>
```

Usage

- The `retrieve()` call *does not* return models in these four states: `Cloning`, `Cloning Failed`, `Deleting`, and `Deletion Failed`.
- Whenever a model is created, its initial state is `Planning`. You can only do a `deploy()` operation for models in `Planning` or `Active` state. The same requirement applies to territories and rules associated with those models. For example, sometimes you can have a model in `Planning` state on a sandbox org, and a model with the same developer name in `Archived` state on your production org. The `deploy()` operation on production fails because that model's state is `Archived` and that state prevents changes to the model.

- Because of the state restrictions, if you have territory models in different orgs with identical developer names and you attempt a `deploy()` operation, Metadata API attempts to create new models. However, that operation fails because of the developer name conflict. For example, sometimes you can have a model in `Planning` state on a sandbox org, and a model with the same developer name in `Archived` state on your production org. The `deploy()` operation on production fails because that model's state is `Archived` and that state prevents changes to the model.
- If you try to delete a model that has territories, then the `delete()` call changes the model's state to `Deleting` and cascade deletes all territories, rules, and user associations in the model. Deleting can take some time depending on the number of territories in the model.
- Whenever a model is created, its initial state is `Planning`. If a model with the same developer name already exists, it already has a state, so we do not include the `State` field in Territory2.
- Sales Territories components don't support packaging or change sets and aren't supported in CRUD calls.
- Namespaces aren't supported for unlocked packages.

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

Territory2Rule

Represents the metadata associated with a territory assignment rule associated with an object, such as Account. Available if Sales Territories has been enabled.

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

Territory2Rule components have the suffix `territory2Rule` and are stored in the `rules` folder under the folder for the corresponding Territory2Model.

Version

Territory2Rule components are available in API version 32.0 and later.

Special Access

The Territory2Model object has a `State` field in SOAP API. States include `Planning`, `Active`, `Archived`, and several other states, such as `Cloning`, that indicate that a process is underway. Users who don't have the Manage Territories permission can access rules that belong to the model in `Active` state. The Manage Territories permission is required for `deploy()` calls for all territory management entities, in addition to the permissions required by Metadata API. Using `retrieve()` without the Manage Territories permission returns only entities that belong to a Territory2Model in `Active` state. We recommend against retrieving without the Manage Territories permission because the call retrieves only partial data.

The SOAP API and the user interface require that a user attempting to create or edit a rule has field-level security access to the fields referenced in the rule item. This restriction is relaxed for Metadata API `deploy()` operations, as they require both Manage Territories and either the Modify Metadata Through Metadata API Functions or Modify All Data permissions.

Fields

Field Name	Field Type	Description
`active`	boolean	Required. Indicates whether the rule is active (`true`) or inactive (`false`). Via the API, active rules run automatically when object records are created and edited. The exception is when the value of the `IsExcludedFromRealign` field on an object record is `true`, which prevents record assignment rules from evaluating that record.
`booleanFilter`	string	An advanced filter condition. For example: `(1 AND 2) OR 3`. Numbering must start at 1 and must be contiguous.
`name`	string	Required. The user interface label for the rule.
`objectType`	string	Required. The object that the rule is defined for. For API version 32.0, the only available object is Account.
`ruleItems`	Territory2RuleItem on page 2218	The items that define a rule's the selection criteria, such as `Billing State equals California`.

Territory2RuleItem

Represents the association of a rule item to a rule. Available in API version 32.0 and later.

Field Name	Field Type	Description
`field`	string	The standard or custom object field that the rule item operates on.
`operation`	FilterOperation (enumeration of type string)	The criterion to apply for the rule item. For example: *`equals`* or *`starts with`*. Valid values are: • `equals` • `notEqual` • `lessThan` • `greaterThan` • `lessOrEqual` • `greaterOrEqual` • `contains` • `notContain` • `startsWith` • `includes` • `excludes` • `within` (`DISTANCE` criteria only)
`value`	string	The field value or values to evaluate. For example: if the field is `Billing ZIP/Postal Code`, a value could be `94105`.

Declarative Metadata Sample Definition

The following example shows the definition of a Territory2RuleItem component.

```
<?xml version="1.0" encoding="UTF-8"?>
<Territory2Rule xmlns="http://soap.sforce.com/2006/04/metadata">
    <label>Northern CA</label>
    <description>To capture northern CA based accounts</description>
    <objectType>Account</objectType>
    <active>True</active>
    <ruleItems>
        <field>BillingZip</field>
        <operation>contains</operation>
        <value><94105,94404,94536/value>
    </ruleItems>
    <ruleItems>
        <field>Industry</field>
        <operation>equals</operation>
        <value>IT</value>
    </ruleItems>
    <ruleItems>
        <field>someCustomField__c</field>
        <operation>greater_than</operation>
        <value>50000</value>
    </ruleItems>
    <booleanFilter>(1 OR 2) AND 3</booleanFilter>
</Territory2Rule>
```

The following is a `package.xml` sample. *`FY13`* and *`FY14`* represent names of territory models and demonstrate that rules can have *identical* developer names within *different* models. A wildcard character (*) in place of the model name can be used to retrieve all rules in all models in an org.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
        <types>
            <members>FY13</members>
            <members>FY14</members>
            <name>Territory2Model</name>
        </types>

        <types>
            <members>FY13.AccRule1</members>
            <members>FY14.AccRule1</members>
            <name>Territory2Rule</name>
        </types>

        <version>64.0</version>
</Package>
```

Usage

- A territory rule can have up to 10 rule items.

- The sort order of rule items is implicitly derived from the position of the rule items in the XML
- Rules can't be run via Metadata API.
- Territory Management 2.0 components don't support packaging or change sets and aren't supported in CRUD calls.

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

Territory2Type

Represents the metadata for a category of territories in Sales Territories. Every Territory2 must have a Territory2Type. This type extends the Metadata metadata type and inherits its `fullName` field.Available if Sales Territories has been enabled.

File Suffix and Directory Location

Territory2Type components have the suffix `territory2Type` and are stored in the `territory2Types` folder.

Version

Territory2Type components are available in API version 32.0 and later.

Special Access Rules

The Manage Territories permission is required for the `deploy()` operation, but not `retrieve()`. The `retrieve()` operation retrieves all the Territory2Type components in the org.

Fields

Field Name	Field Type	Description
`description`	string	A description of the territory type.
`name`	string	Required. The user interface label for the territory type.
`priority`	int	Required. Used for Filter-Based Opportunity Territory Assignment (Pilot in Spring '15 / Metadata API version 33). Lets you specify a priority for a territory type. For opportunity assignments, the filter examines all territories assigned to the account that the opportunity is assigned to. The account-assigned territory whose territory type priority is highest is then assigned to the opportunity. The `priority` field value on each territory type must be unique. Further, if there are multiple territories with the same territory type, and therefore the same priority, assigned to the account, no territory is not assigned to the opportunity.

Declarative Metadata Sample Definition

The following example shows the definition of a Territory2Type component.

```
<?xml version="1.0" encoding="UTF-8"?>
<Territory2Type xmlns="http://soap.sforce.com/2006/04/metadata">
    <name>Geo</name>
    <description>Geographic allocation</description>
</Territory2Type>
```

Usage

Sales Territories components don't support packaging or change sets and aren't supported in CRUD calls.

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

TimelineObjectDefinition

Represents the container that stores the details of a timeline configuration. You can use this resource with Salesforce objects to see their records' related events in a linear time-sorted view.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

TimelineObjectDefinition components have the suffix `.timelineObjectDefinition` and are stored in the `timelineObjectDefinitions` folder.

Version

TimelineObjectDefinition components are available in API version 55.0 and later.

Special Access Rules

TimelineObjectDefinition is available in any org that has the Timeline org preference enabled.

Fields

Field Name	Description
`baseObject`	**Field Type** string **Description** Required. The object on which a timeline is based. Information displayed in a timeline comes from objects that are related to the base object. The base object can be a Salesforce object or custom object.
`definition`	**Field Type** string **Description** Required. The timeline definition in JSON format.
`isActive`	**Field Type** boolean **Description** Indicates whether the timeline is active (`true`) or not (`false`).
`masterLabel`	**Field Type** string **Description** Required. The user interface label of the timeline object definition record.

Declarative Metadata Sample Definition

The following is an example of a TimelineObjectDefinition component.

```
<?xml version="1.0" encoding="UTF-8"?>
<TimelineObjectDefinition
 xmlns="http://soap.sforce.com/2006/04/metadata">
 <baseObject>Account</baseObject>

<definition>{&quot;timelineObjects&quot;:[{&quot;object&quot;:{&quot;type&quot;:&quot;Account&quot;,&quot;label&quot;:&quot;Account&quot;,&quot;name&quot;:&quot;Account&quot;},&quot;name&quot;:{&quot;field&quot;:&quot;Name&quot;,&quot;label&quot;:&quot;Name&quot;,&quot;type&quot;:&quot;Auto

Number&quot;,&quot;type&quot;:&quot;AutoNumber&quot;},&quot;subTitle&quot;:{&quot;field&quot;:&quot;Comments&quot;,&quot;label&quot;:&quot;Internal
```

```
Date&quot;,&quot;type&quot;:&quot;DateTime&quot;},&quot;anchorReferenceField&quot;:{&quot;field&quot;:&quot;AccountId&quot;,&quot;label&quot;:&quot;Account
 Date
Only&quot;,&quot;type&quot;:&quot;DueDate&quot;},&quot;anchorReferenceField&quot;:{&quot;field&quot;:&quot;WhatId&quot;,&quot;label&quot;:&quot;Related
 To
Result&quot;,&quot;type&quot;:&quot;Text&quot;},{&quot;field&quot;:&quot;CallObject&quot;,&quot;label&quot;:&quot;Call
 Object
Identifier&quot;,&quot;type&quot;:&quot;Text&quot;},{&quot;field&quot;:&quot;CallType&quot;,&quot;label&quot;:&quot;Call
 Date
Only&quot;,&quot;type&quot;:&quot;DueDate&quot;},&quot;anchorReferenceField&quot;:{&quot;field&quot;:&quot;WhatId&quot;,&quot;label&quot;:&quot;Related
 To ID&quot;,&quot;type&quot;:&quot;EntityId&quot;}}}]}}</definition>
 <isActive>true</isActive>
 <masterLabel>HealthTimeline</masterLabel>
</TimelineObjectDefinition>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package
 xmlns="http://soap.sforce.com/2006/04/metadata">
 <types>
  <members>*</members>
  <name>TimelineObjectDefinition</name>
 </types>
 <version>55.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

TimeSheetTemplate

Represents a template for creating time sheets in Field Service. This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

TimeSheetTemplate components have the suffix `timeSheetTemplate` and are stored in the `timeSheetTemplates` folder.

Version

TimeSheetTemplate components are available in API version 46.0 and later.

Special Access Rules

Field Service must be enabled. Users must have the Customize Application and Time Sheet Template permissions.

Fields

Field Name	Field Type	Description
`active`	boolean	Required. Indicates whether the time sheet template is active (`true`) or not (`false`).
`description`	string	The time sheet template's description.
`frequency`	TimeSheetFrequency (enumeration of type string)	Required. Defines the frequency of the time sheet creation period. One of the following values: • `Daily` • `Weekly` • `EveryTwoWeeks` • `TwiceAMonth` • `Monthly`
`masterLabel`	string	Required. The name of the time sheet template.
`startDate`	date	Required. The date when the time sheet takes effect.
`timeSheetTemplateAssignments`	TimeSheetTemplateAssignment	A list of profiles that the template is assigned to.
`workWeekEndDay`	DaysOfWeek (enumeration of type string)	Required. The end day of the template's work week. One of the following values: • `Monday` • `Tuesday` • `Wednesday` • `Thursday` • `Friday` • `Saturday` • `Sunday`

Field Name	Field Type	Description
workWeekStartDay	DaysOfWeek (enumeration of type string)	Required. The start day of the template's work week. One of the following values: • Monday • Tuesday • Wednesday • Thursday • Friday • Saturday • Sunday

TimeSheetTemplateAssignment

Returns a quick action that's associated with an EmbeddedServiceLiveAgent setup. The quick action includes the pre-chat form fields that the embedded chat window displays and shows the order in which the fields are displayed.

Field Name	Field Type	Description
assignedTo	string	The IDs of the user profiles that a time sheet template is assigned to.

Declarative Metadata Sample Definition

The following is an example of a TimeSheetTemplate file.

```
<?xml version="1.0" encoding="UTF-8"?>
<TimeSheetTemplate xmlns="http://soap.sforce.com/2006/04/metadata">
   <active>true</active>
   <description>Time Sheet Template description</description>
   <frequency>Daily</frequency>
   <masterLabel>label</masterLabel>
   <startDate>2018-10-18</startDate>
   <timeSheetTemplateAssignments>
       <assignedTo>admin</assignedTo>
   </timeSheetTemplateAssignments>
   <timeSheetTemplateAssignments>
       <assignedTo>standard</assignedTo>
   </timeSheetTemplateAssignments>
   <workWeekEndDay>Tuesday</workWeekEndDay>
   <workWeekStartDay>Monday</workWeekStartDay>
</TimeSheetTemplate>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
   <types>
       <members>*</members>
       <name>TimeSheetTemplate</name>
```

```
    </types>
    <version>46.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

TopicsForObjects

Represents the ability to assign topics to objects or to remove topic assignments.

File Suffix and Directory Location

TopicsForObjects components have the suffix `.topicsForObjects` and are stored in the `topicsForObjects` folder of the corresponding package directory.

Version

TopicsForObjects components are available in API version 41.0 and later.

Fields

Field Name	Field Type	Description
`enableTopics`	boolean	Required. When true, indicates whether users can assign topics or remove topic assignments. When false, users can't assign or remove topics. Upon org creation, this value is true for the following objects: • Account • Asset • Campaign • Case • Contact • Content Document • Contract • Event • Lead • Opportunity • Order • Solution • Task

Field Name	Field Type	Description
		For all remaining standard objects and custom objects, the default is false.
`entityApiName`	string	Required. Indicates the object's API name for enabling topics.

Declarative Metadata Sample Definition

The following is an example of a TopicsForObjects component.

```
<?xml version="1.0" encoding="UTF-8"?>
<TopicsForObjects xmlns="http://soap.sforce.com/2006/04/metadata">
       <enableTopics>false</enableTopics>
       <entityApiName>Account</entityApiName>
</TopicsForObjects>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
<types>
<members>*</members>
<name>TopicsForObjects</name>
</types>
<version>41.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

TransactionSecurityPolicy

Represents a transaction security policy definition. Transaction security policies give you a way to look through events in your organization and specify actions to take when certain combinations occur.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

TransactionSecurityPolicy components have the suffix `.transactionSecurityPolicy` and are stored in the `transactionSecurityPolicies` folder.

Version

TransactionSecurityPolicy components are available in API version 35.0 and later.

Fields

Field Name	Field Type	Description
`action`	TransactionSecurityAction	Required. Describes the action to take when the matching Transaction Security policy is triggered.
`active`	boolean	Required. If `true`, the policy is enabled and actively monitors its event.
`apexClass`	string	Required for Apex-based policies, and optional for all other policies. The name of the class that implements the `TxnSecurity.PolicyCondition` or `TxnSecurity.EventCondition` interface for this policy. Available in API version 46.0 and later.
`blockMessage`	string	The custom message sent to a user when a policy blocks their action. Used in Real-Time Event Monitoring only. Maximum of 1000 characters. This field is null when the default message option is selected in the UI. Available only when `eventName` is set to `ApiEvent`, `ListViewEvent`, `BulkApiResultEventStore`, or `ReportEvent`. Available in API version 49.0 and later. Include org- or policy-specific information in your custom message, such as the name of the responsible administrator or the business unit. Be careful about what you include. Too much information on how the policy was designed. can aid a malicious user. Two-factor authentication (2FA) isn't supported in Lightning Experience, so events like `ListView` and `ReportEvent` are upgraded to Block in Lightning. Custom messages aren't translatable.
`customEmailContent`	string	The administrator-created custom email content sent when a policy is triggered. Used in Real-Time Event Monitoring only. Maximum of 1333 characters. This field is null when the Custom Email Content setting is selected in the UI but no message content is entered. Available in API version 54.0 and later. Custom messages aren't translatable.
`description`	string	A description of the policy.
`developerName`	string	This unique name prevents conflicts with other policies that have the same `masterLabel`. This name can contain only underscores and alphanumeric characters, and must be unique in

Field Name	Field Type	Description
		your org. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. Only users with View DeveloperName OR View Setup and Configuration permission can view, group, sort, and filter this field.
`eventName`	TransactionSecurityEventName (enumeration of type string)	Used in Real-Time Event Monitoring only. Indicates the name of the event the policy monitors. This field is available in API 45.0 and later. Valid values are: • `ApiEvent`—Tracks these user-initiated read-only API calls: `query()`, `queryMore()`, and `count()`. Captures API requests through SOAP API and Bulk API for the Enterprise and Partner WSDLs. Tooling API calls and API calls originating from a Salesforce mobile app aren't captured. • `ApiAnomalyEventStore`—Tracks anomalies in how users make API calls. ApiAnomalyEventStore is an object that stores the event data of ApiAnomalyEvent. This object is available in API version 50.0 and later. • `BulkApiResultEventStore`—Tracks when a user downloads the results of a Bulk API request. BulkApiResultEventStore is a big object that stores the event data of BulkApiResultEvent. This object is available in API version 50.0 and later. • `CredentialStuffingEventStore`—Tracks when a user successfully logs into Salesforce during an identified credential stuffing attack. Credential stuffing refers to large-scale automated login requests using stolen user credentials.This value is available in API version 49.0 and later. • `FileEventStore` (beta)—Tracks when a user downloads, previews, or uploads a file. FileEventStore is a big object that stores the event data of FileEvent. This object is available in API version 57.0 and later. • `GuestUserAnomalyEventStore`—Tracks data access anomalies that are caused by guest user permission misconfiguration. This object is available in API version 60.0 and later. • `ListViewEvent`—Tracks when users access data with list views using Lightning Experience, Salesforce Classic, or the API. It doesn't track list views of Setup entities. • `LoginAsEvent`—Tracks the login activity of admins who log in to Salesforce as other users. This object is available in API version 46.0 and later. • `LoginEvent`—LoginEvent tracks the login activity of users who log in to Salesforce.

Field Name	Field Type	Description
		• `PermissionSetEventStore` —Tracks changes to permission sets and permission set groups. • `ReportAnomalyEventStore`—Tracks anomalies in how users run or export reports, including unsaved reports. This value is available in API version 49.0 and later. • `ReportEvent`—Tracks when reports are run in your org. • `SessionHijackingEventStore`—Tracks when unauthorized users gain ownership of a Salesforce user's session with a stolen session identifier. To detect such an event, Salesforce evaluates how significantly a user's current browser fingerprint diverges from the previously known fingerprint using a probabilistically inferred significance of change. Available in API version 49.0 and later.
`eventType`	MonitoredEvents (enumeration of type string)	Used in Legacy Transaction Security only. Required for Apex-based policies, and optional for all other policies. Indicates which type of event is being monitored. Valid values are: • `AccessResource`—Notifies you when the selected resource has been accessed. • `AuditTrail`—Reserved for future use. • `DataExport`—Notifies you when the selected object type has been exported using the Data Loader API client. • `Entity`—Notifies you on use of an object type such as an authentication provider or Chatter comment. • `Login`—Notifies you when a user logs in. As of Summer '20, Legacy Transaction Security is a retired feature in all Salesforce orgs.
`executionUser`	string	Used in Legacy Transaction Security only. The name or ID of an active user who is assigned the Modify All Data and View Setup user permissions. As of Summer '20, Legacy Transaction Security is a retired feature in all Salesforce orgs.
`flow`	string	Required only for policies of type `CustomConditionBuilderPolicy`. The ID of the Flow object that contains the logic the Condition Builder transaction security policy. Available in API version 46.0 and later.
`masterLabel`	string	The label for this object. This display value is the internal label that is't translated. Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Field Name	Field Type	Description
`resourceName`	string	Used in Legacy Transaction Security only. Required for Apex-based policies, and optional for all other policies. A resource used to narrow down the conditions under which the policy triggers. For example, with a `DataExport` event, you can select a resource Lead to specifically monitor export activity occurring on your Lead entities. The resources available depend on the `Event Type` field. The following valid resources are grouped by event type. • AccessResource—ConnectedApplication, Reports • DataExport—Account, Case, Contact, Lead, Opportunity • Entity—AuthProvider, ChatterMessage, FeedComment, FeedItem, Idea, Question • Login—LoginHistory As of Summer '20, Legacy Transaction Security is a retired feature in all Salesforce orgs.
`type`	TxnSecurityPolicyType (enumeration of type string)	The type of validation that the policy uses. The valid values are: • `CustomApexPolicy`— Created with Apex editor. • `CustomConditionBuilderPolicy`— Created with Condition Builder. The default value is `CustomApexPolicy`.

TransactionSecurityAction

Describes the action to take when the matching Transaction Security policy is triggered.

Field Name	Field Type	Description
`block`	boolean	If `true`, the requested operation is blocked. This action only applies to Login and AccessResource events.
`endSession`	boolean	Used in Legacy Transaction Security only. If `true`, a current session must be closed before a new session can be started. This action only applies to Login events. As of Summer '20, Legacy Transaction Security is a retired feature in all Salesforce orgs.
`freezeUser`	boolean	Used in Legacy Transaction Security only. If `true`, the user that triggered the policy is frozen. This action only applies to Chatter resources for Entity events. As of Summer '20, Legacy Transaction Security is a retired feature in all Salesforce orgs.
`notifications`	TransactionSecurityNotification[]	Specifies how to notify the Salesforce administrator when the action is triggered. There can be none, one, or multiple notifications.

Field Name	Field Type	Description
twoFactorAuthentication	boolean	If true, multi-factor authentication (MFA) is required for a higher level of access before the requested operation can continue. This action only applies to Login and AccessResource events. Multi-factor authentication was formerly called two-factor authentication.

TransactionSecurityNotification

Describes who to notify and how to notify them when the matching Transaction Security policy is triggered.

Field Name	Field Type	Description
inApp	boolean	True if an in-app notification is selected.
sendEmail	boolean	True if an email notification is selected.
user	string	The user to receive the notification.

Declarative Metadata Sample Definition

The following is an example of a Real-Time Event Monitoring TransactionSecurityPolicy component.

```
<?xml version="1.0" encoding="UTF-8"?>
<TransactionSecurityPolicy xmlns="http://soap.sforce.com/2006/04/metadata">
    <action>
        <block>true</block>
        <notifications>
            <inApp>true</inApp>
            <sendEmail>true</sendEmail>
            <user>user@your.org</user>
        </notifications>
        <twoFactorAuthentication>false</twoFactorAuthentication>
    </action>
    <active>true</active>
    <apexClass>TxnSecMDApiPolicyEventCondition</apexClass>
    <blockMessage>You cannot view this report.</blockMessage>
    <developerName>TxnSecPolicyMDApi</developerName>
    <eventName>ReportEvent</eventName>
    <masterLabel>Txn Sec MD Api Policy</masterLabel>
    <type>CustomApexPolicy</type>
</TransactionSecurityPolicy>
```

The following is an example package manifest used to deploy or retrieve the transaction security metadata for an organization.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>MySecurityPolicy</members>
        <name>TransactionSecurityPolicy</name>
```

```
    </types>
    <version>35.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

Translations

Metadata type that enables work with translations for various supported languages. The ability to translate component labels is part of the Translation Workbench.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

This type extends the Metadata metadata type and inherits its `fullName` field.

Language

A two-character language code identifies each language, such as `en`. A five-character code is used for languages that differ depending on location. For example, en_AU.

Note: Setting a default language is different from setting a default locale. For more information, see Select Your Language, Locale, and Currency in Salesforce Help.

Salesforce offers full support for these languages.

- Chinese (Simplified): `zh_CN`
- Chinese (Traditional): `zh_TW`
- Danish: `da`
- Dutch: `nl_NL`
- English: `en_US`
- Finnish: `fi`
- French: `fr`
- German: `de`
- Italian: `it`
- Japanese: `ja`
- Korean: `ko`
- Norwegian: `no`
- Portuguese (Brazil): `pt_BR`
- Russian: `ru`
- Spanish: `es`
- Spanish (Mexico): `es_MX` Spanish (Mexico) defaults to Spanish for customer-defined translations.
- Swedish: `sv`

- Thai: `th` The Salesforce user interface is fully translated to Thai, but Help is in English.

End-user languages are useful if you have a multilingual organization or partners who speak languages other than your company's default language. For end-user languages, Salesforce provides translated labels for standard objects and pages, except admin pages, Setup, and Help. Some clouds and features support a subset of these languages in the UI. For details, see User Interface Language Support in Salesforce Help. When you select an end-user language, labels that aren't translated and Salesforce Help appear in English. End-user languages are intended only for personal use by end users. Don't use end-user languages as corporate languages. Salesforce doesn't provide customer support in end-user languages.

End-user languages include:

- Arabic: `ar`
- Bulgarian: `bg`
- Croatian: `hr`
- Czech: `cs`
- English (UK): `en_GB`
- Greek: `el`
- Hebrew: `iw`
- Hungarian: `hu`
- Indonesian: `in`
- Polish: `pl`
- Portuguese (European): `pt_PT`
- Romanian: `ro`
- Slovak: `sk`
- Slovenian: `sl`
- Turkish: `tr`
- Ukrainian: `uk`
- Vietnamese: `vi`

Important: Before enabling end-user languages Arabic and Hebrew, review the right-to-left language support limitations.

In situations where Salesforce doesn't provide default translations, use platform-only languages to localize apps and custom functionality that you build on the Salesforce Platform. You can translate items such as custom labels, custom objects, and field names. You can also rename most standard objects, labels, and fields. Informational text and non-field label text aren't translatable.

Platform-only languages are available in all places where you can select a language in the application. However, when you select a platform-only language, all standard Salesforce labels default to English or, in select cases, to an end-user or fully supported language.

Note: Language support is closely tied to the API version. For example, we introduced support for Belgian Dutch (nl_BE) in API version 40.0. To take advantage of this language, you must use API version 40.0 or later. In general, we recommend using the most recent version of the API to make the most of our language features.

Platform-only languages include:

- Albanian: `sq`
- Afrikaans: `af`
- Amharic: `am`
- Arabic (Algeria): `ar_DZ`
- Arabic (Bahrain): `ar_BH`

- Arabic (Egypt): `ar_EG`
- Arabic (Iraq): `ar_IQ`
- Arabic (Jordan): `ar_JO`
- Arabic (Kuwait): `ar_KW`
- Arabic (Lebanon): `ar_LB`
- Arabic (Libya): `ar_LY`
- Arabic (Morocco): `ar_MA`
- Arabic (Oman): `ar_OM`
- Arabic (Qatar): `ar_QA`
- Arabic (Saudi Arabia): `ar_SA`
- Arabic (Sudan): `ar_SD`
- Arabic (Syria): `ar_SY`
- Arabic (Tunisia): `ar_TN`
- Arabic (United Arab Emirates): `ar_AE`
- Arabic (Yemen): `ar_YE`
- Armenian: `hy`
- Basque: `eu`
- Bosnian: `bs`
- Bengali: `bn`
- Burmese: `my`
- Catalan: `ca`
- Chinese (Hong Kong): `zh_HK`
- Chinese (Singapore): `zh_SG`
- Chinese (Malaysia): `zh_MY`
- Dutch (Belgium): `nl_BE`
- English (Australia): `en_AU`
- English (Belgium): `en_BE`
- English (Canada): `en_CA`
- English (Cyprus): `en_CY`
- English (Germany): `en_DE`
- English (Hong Kong): `en_HK`
- English (India): `en_IN`
- English (Ireland): `en_IE`
- English (Israel): `en_IL`
- English (Malaysia): `en_MY`
- English (Malta): `en_MT`
- English (Netherlands): `en_NL`
- English (New Zealand): `en_NZ`
- English (Philippines): `en_PH`
- English (Singapore): `en_SG`

- English (South Africa): `en_ZA`
- English (United Arab Emirates): `en_AE`
- Estonian: `et`
- Farsi: `fa`
- French (Belgium): `fr_BE`
- French (Canada): `fr_CA`
- French (Luxembourg): `fr_LU`
- French (Morocco): `fr_MA`
- French (Switzerland): `fr_CH`
- Georgian: `ka`
- German (Austria): `de_AT`
- German (Belgium): `de_BE`
- German (Luxembourg): `de_LU`
- German (Switzerland): `de_CH`
- Greek (Cyprus): `el_CY`
- Greenlandic: `kl`
- Gujarati: `gu`
- Hawaiian: `haw`
- Haitian Creole: `ht`
- Hindi: `hi`
- Hmong: `hmn`
- Icelandic: `is`
- Irish: `ga`
- Italian (Switzerland): `it_CH`
- Kannada: `kn`
- Kazakh: `kk`
- Khmer: `km`
- Latvian: `lv`
- Lithuanian: `lt`
- Luxembourgish: `lb`
- Macedonian: `mk`
- Malay: `ms`
- Malayalam: `ml`
- Maltese: `mt`
- Marathi: `mr`
- Montenegrin: `sh_ME`
- Punjabi: `pa`
- Romanian (Moldova): `ro_MD`
- Romansh: `rm`
- Russian (Armenia): `ru_AM`

- Russian (Belarus): `ru_BY`
- Russian (Kazakhstan): `ru_KZ`
- Russian (Kyrgyzstan): `ru_KG`
- Russian (Lithuania): `ru_LT`
- Russian (Moldova): `ru_MD`
- Russian (Poland): `ru_PL`
- Russian (Ukraine): `ru_UA`
- Samoan: `sm`
- Serbian (Cyrillic): `sr`
- Serbian (Latin): `sh`
- Spanish (Argentina): `es_AR`
- Spanish (Bolivia): `es_BO`
- Spanish (Chile): `es_CL`
- Spanish (Colombia): `es_CO`
- Spanish (Costa Rica): `es_CR`
- Spanish (Dominican Republic): `es_DO`
- Spanish (Ecuador): `es_EC`
- Spanish (El Salvador): `es_SV`
- Spanish (Guatemala): `es_GT`
- Spanish (Honduras): `es_HN`
- Spanish (Nicaragua): `es_NI`
- Spanish (Panama): `es_PA`
- Spanish (Paraguay): `es_PY`
- Spanish (Peru): `es_PE`
- Spanish (Puerto Rico): `es_PR`
- Spanish (United States): `es_US`
- Spanish (Uruguay): `es_UY`
- Spanish (Venezuela): `es_VE`
- Swahili: `sw`
- Tagalog: `tl`
- Tamil: `ta`
- Te reo: `mi`
- Telugu: `te`
- Urdu: `ur`
- Welsh: `cy`
- Xhosa: `xh`
- Yiddish: `ji`
- Zulu: `zu`

Important: Before enabling Urdu as a platform-only language, review the right-to-left language support limitations.

Declarative Metadata File Suffix and Directory Location

Local translations are stored in a file with a format of *localeCode*.translation, where *localeCode* is the locale code of the translation language. For example, the file name for German translations is de.translation. Packaged translations are stored in a file with a format of *pkgNamespace*_ _*localeCode*.translation. For example, if the package namespace is Acme, the file name for German translations installed by the package is Acme_ _de.translation. The supported locale codes are listed in Language.

Custom object translations are stored in the objectTranslations folder in the corresponding package directory.

Version

Translations components are available in API version 14.0 and later.

Fields

Field	Field Type	Description
bots	BotTranslation[]	A list of bot translations. Available in API version 53.0 and later.
customApplications	CustomApplicationTranslation[]	A list of custom application translations.
customLabels	CustomLabelTranslation[]	A list of custom label translations.
customPageWebLinks	CustomPageWebLinkTranslation[]	A list of translations for web links defined in a home page component.
customTabs	CustomTabTranslation[]	A list of custom tab translations.
desFieldTemplateMessages	ExplainabilityMsgTemplateFieldTranslation[]	A list of admin-configured explainability message templates.
flowDefinitions	FlowDefinitionTranslation[]	A list of flow translations. Only Flow and AutolaunchedFlow types are supported for translation. This field is available in API version 41.0 and later.
identityVerificationCustomFieldLabels	IdentityVerificationFieldTranslation	A list of identity verification translation fields. This field is available in API version 54.0 and later.
fullName	string	Required. The language code. For example, de for German. Inherited from Metadata, this field is defined in the WSDL for this metadata type. It must be specified when creating, updating, or deleting. For an example of this field specified for a call, see createMetadata().

Field	Field Type	Description
`globalPicklists`	GlobalPicklistTranslation[]	A list of global picklist translations. A global picklist's value set is inherited by all the custom picklist fields that are based on it. This field is available in API version 37.0 only and is removed from later versions.
`pipelineInspMetricConfigs`	PipelineInspMetricConfigTranslation	A list of translations of Pipeline Inspection forecast category metric settings. This field is available in API version 57.0 and later.
`prompts`	PromptTranslation	A list of In-App Guidance prompt translations. This field is available in API version 48.0 and later.
`quickActions`	GlobalQuickActionTranslation[]	A list of global rather than object-specific quick actions.
`reportTypes`	ReportTypeTranslation[]	A list of report type translations.
`scontrols`	ScontrolTranslation[]	A list of s-control translations.

BotTranslation

BotTranslation contains details for a translation of a bot. Available in API version 53.0 and later.

Field	Field Type	Description
`botVersions`	BotVersionTranslation[]	A list of bot version translations.
`fullName`	string	Required. Name of the bot. The `fullName` for the translation must match the `fullName` inherited by the original Bot type.

BotVersionTranslation

BotVersionTranslation contains details for a translation of a bot version. Available in API version 53.0 and later.

Field	Field Type	Description
`botDialogs`	BotDialogTranslation[]	A translated list of dialogs in this bot version.
`fullName`	string	Required. Name of a bot version. The `fullName` for the translation must match the `fullName` inherited by the original BotVersion type.

BotDialogTranslation

BotDialogTranslation contains details for a translation of a bot dialog. Available in API version 53.0 and later.

Field	Field Type	Description
`botSteps`	BotStepTranslation[]	A translated list of steps that are executed as part of the dialog.
`developerName`	string	Required. This unique name prevents conflicts with other dialogs associated with the same bot version. The `developerName` for the translation must match the `developerName` on the original BotDialog subtype of BotVersion.
`label`	string	A translated label that identifies the dialog throughout the Salesforce user interface.

Note: In Metadata Deployment of Translations, it's expected that blank values cannot be used to delete existing translations. If a translation label is left blank, it's skipped during deployment, and no error will be shown.

BotStepTranslation

BotStepTranslation contains details for a translation of a bot step. Available in API version 53.0 and later.

Field	Field Type	Description
`botMessages`	BotMessageTranslation[]	A translated list of bot messages used by a BotStep of type `Message`.
`botSteps`	BotStepTranslation[]	A translated list of bot steps associated with a BotStep of type `Group`.
`botVariableOperation`	BotVariableOperationTranslation	A translated bot variable operation used by a BotStep of type `VariableOperation`.
`stepIdentifier`	string	Required. A unique key that identifies a step within a dialog. This key is used to link translated labels to labels within the step. This field is recommended for all step records and is required for translated step labels. The `stepIdentifier` for the translation must match the `stepIdentifier` on the original BotStep subtype of BotVersion.
`type`	BotStepType (enumeration of type string)	Required. Valid values are: • `Navigation` • `Invocation` • `VariableOperation` • `Message` • `Wait` • `Group` • `RecordLookup` (Available in API version 48.0 and later.)

Field	Field Type	Description
		The `type` for the translation must match the `type` on the original BotStep subtype of BotVersion.

BotMessageTranslation

BotMessageTranslation contains details for a translation of a bot message step. Available in API version 53.0 and later.

Field	Field Type	Description
`message`	string	A translated message to display as part of an outgoing message from the bot to the customer.
`messageIdentifier`	string	Required. A unique key that identifies a message within a dialog. This key is used to link translated labels to labels within the message. This field is recommended for all message records and is required for translated message labels. The `messageIdentifier` for the translation must match the `messageIdentifier` on the original BotMessage subtype of BotVersion.

BotVariableOperationTranslation

BotVariableOperationTranslation contains details for a translation of a bot variable operation (question) step. Available in API version 53.0 and later.

Field	Field Type	Description
`botMessages`	BotMessageTranslation on page 2241[]	A translated list of bot messages used as prompt messages by a BotVariableOperation of type `Collect`.
`botQuickReplyOptions`	BotQuickReplyOptionTranslation on page 2242[]	A translated list of static choice options used by a BotVariableOperation of type `Collect` and `quickReplyType` of `Static`.
`quickReplyOptionTemplate`	string	A translated formula template used to resolve a label for Dynamic choice options of type `Object`.
`retryMessages`	BotMessageTranslation on page 2241[]	In Conversation Repair, the translated messages assigned to repair attempts.
`successMessages`	BotMessageTranslation on page 2241[]	In a File dialog step, the translated message displayed to the customer as part of type CollectAttachment to confirm a successful file upload. Available in API version 57.0 and later.
`type`	BotVariableOperationType (enumeration of type string)	Required. Valid values are: • `Set` • `Unset`

Field	Field Type	Description
		• `Collect` • `SetConversationLanguage`
`variableOperationIdentifier`	string	Required. A unique key that identifies a variable operation within a dialog. This key is used to link translated labels to labels within the variable operation. This field is recommended for all variable operation records and is required for translated variable operation labels. The `variableOperationIdentifier` for the translation must match the `variableOperationIdentifier` on the original BotVariableOperation subtype of BotVersion.

BotQuickReplyOptionTranslation

BotQuickReplyOptionTranslation contains details for a translation of a bot quick reply option within a bot variable operation (question) step. Available in API version 53.0 and later.

Field	Field Type	Description
`literalValue`	string	A translated value to be displayed as a menu or button choice to your customer.
`quickReplyOptionIdentifier`	string	Required. A unique key that identifies a quick reply option within a dialog. This key is used to link translated labels to labels within the quick reply option. This field is recommended for all quick reply option records and is required for translated quick reply option labels. The `quickReplyOptionIdentifier` for the translation must match the `quickReplyOptionIdentifier` on the original BotQuickReplyOption subtype of BotVersion.

CustomApplicationTranslation

CustomApplicationTranslation contains details for a custom application translation. For more details, see CustomApplication.

Field	Field Type	Description
`description`	string	The optional description text of the application translation.
`label`	string	Required. The translated custom application name. Maximum of 765 characters.
`name`	string	Required. The name of the custom application.

CustomLabelTranslation

CustomLabelTranslation contains details for a custom label translation. For more details, see CustomLabels.

Field	Field Type	Description
label	string	Required. The translated custom label name. Maximum of 765 characters.
name	string	Required. The custom label name.

CustomPageWebLinkTranslation

CustomPageWebLinkTranslation contains details for a translation of a web link defined in a home page component. For more details, see CustomPageWebLink.

Field	Field Type	Description
label	string	Required. The translated web link.
name	string	Required. The name of the web link.

CustomTabTranslation

CustomTabTranslation contains details for a translation of a custom tab. For more details, see CustomTab.

Field	Field Type	Description
label	string	Required. The translated custom tab name.
name	string	Required. The custom tab name.

ExplainabilityMsgTemplateFieldTranslation

Represents the template that contains the decision explanation message for a specified step element type.

Field Name	Field Type	Description
description	string	The explainability message field description.
label	string	A user-friendly name for ExplainabilityMsgTemplateFieldTranslation.
name	string	Required. The name of the decision explanation message for a specified step element type.
templateMessage	string	The message associated with the template for a specific expression set step type.

Declarative Metadata Sample Definition

This is an example of an ExplainabilityMsgTemplateFieldTranslation component.

```
<?xml version="1.0" encoding="UTF-8"?>
<Translations xmlns="http://soap.sforce.com/2006/04/metadata">
    <desFieldTemplateMessages>
        <description>Calc Blitz Message</description>
        <label>CALBLITZ</label>
        <name>CALBLITZ</name>
        <templateMessage>CALBLITZ</templateMessage>
    </desFieldTemplateMessages>
</Translations>
```

FlowDefinitionTranslation

FlowDefinitionTranslation contains details for a translation of a flow definition. For more details, see FlowDefinition.

Available in API version 41.0 and later.

Field	Field Type	Description
`flows`	FlowTranslation[]	A list of flow version translations for the flow definition.
`fullName`	string	Required. The API name for the flow definition.
`label`	string	A translated label for the flow definition. By default, flow definitions inherit the label of the active flow version. If you provide a label here, the definition label no longer inherits changes to the active version label.

FlowTranslation

FlowTranslation contains details for a translation of a flow version. For more details, see Flow.

Available in API version 41.0 and later.

Field	Field Type	Description
`choices`	FlowChoiceTranslation[]	A list of choice translations for the flow version.
`fullName`	string	The API name for the flow version. A unique name for the flow that contains only underscores and alphanumeric characters. The name must be unique across the org, begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. To deploy or retrieve a version, you can specify the version number. For example, `sampleFlow-3` specifies version 3 of the flow whose unique name is sampleFlow. If you don't specify a version number, the flow is the latest version.

Field	Field Type	Description
		In API version 43.0 and earlier, this field included the version number. In API version 44 and later, this field no longer includes the version number.
`label`	string	A translated label for the flow version.
`orchestrationSteps`	FlowOrchestrationStageTranslation on page 2246	A list of orchestration step translations for the flow version. Available in API version 63.0 and later.
`orchestrationSteps`	FlowOrchestrationStepTranslation on page 2246	A list of orchestration step translations for the flow version. Available in API version 63.0 and later.
`screens`	FlowScreenTranslation[]	A list of screen translations for the flow version.
`stages`	FlowStageTranslation on page 2248[]	A list of stage translations for the flow version. Available in API version 43.0 and later.

FlowChoiceTranslation

FlowChoiceTranslation contains details for a translation of a choice in a flow version. For more details, see FlowChoice in Flow.

Available in API version 41.0 and later.

Field	Field Type	Description
`choiceText`	string	A translated label for the choice.
`name`	string	Required. A unique name for the choice.
`userInput`	FlowChoiceUserInputTranslation	A translated choice input for the choice.

FlowChoiceUserInputTranslation

FlowChoiceUserInputTranslation contains details for a translation of a choice input. For more details, see FlowChoiceUserInput in Flow.

Available in API version 41.0 and later.

Field	Field Type	Description
`promptText`	string	A translated label for the choice input.
`validationRule`	FlowInputValidationRuleTranslation	A translated validation rule for the choice input.

FlowInputValidationRuleTranslation

FlowInputValidationRuleTranslation contains details for a translation of a validation rule. For more details, see FlowInputValidationRule in Flow.

Available in API version 41.0 and later.

Field	Field Type	Description
errorMessage	string	A translated error message for the validation rule.

FlowOrchestrationStageTranslation

FlowOrchestrationStageTranslation contains details for a translation of an orchestration stage in an orchestration version. For more details, see FlowOrchestratedStage in Flow.

Available in API version 64.0 and later.

Field	Field Type	Description
stageLabel	string	A translated label for the orchestration stage.

FlowOrchestrationStepTranslation

FlowOrchestrationStepTranslation contains details for a translation of an orchestration step in an orchestration version. For more details, see FlowStageStep in Flow.

Available in API version 63.0 and later.

Field	Field Type	Description
stepLabel	string	A translated label for the orchestration step.

FlowScreenTranslation

FlowScreenTranslation contains details for a translation of a screen. For more details, see FlowScreen in Flow.

Available in API version 41.0 and later.

Field	Field Type	Description
backButtonLabel	string	A translated label for the Back button. Available in API version 54.0 and later.
fields	FlowScreenFieldTranslation[]	A list of screen component translations for the screen.
helpText	string	Translated help text for the screen.
name	string	Required. An API name for the screen.
nextOrFinishButtonLabel	string	A translated label for the Next or Finish button. Available in API version 54.0 and later.
pauseButtonLabel	string	A translated label for the Pause button. Available in API version 54.0 and later.
pausedText	string	A translated pause confirmation message for the screen.

FlowScreenFieldTranslation

FlowScreenFieldTranslation contains details for a translation of a screen component. For more details, see FlowScreenField in Flow.

Available in API version 41.0 and later.

Note: Translation isn't supported for screen components that require Lightning runtime.

Field	Field Type	Description
`fieldText`	string	A translated label for the screen component.
`helpText`	string	Translated help text for the screen component.
`inputParameters`	FlowInputParameterTranslation	Reserved for internal use.
`name`	string	Required. An API name for the screen component.
`validationRule`	FlowInputValidationRuleTranslation	Translated validation rule for the screen component.

FlowInputParameterTranslation

FlowInputParameterTranslation is reserved for internal use.

Field	Field Type	Description
`name`	string	Reserved for internal use.
`value`	FlowFerovTranslation	Reserved for internal use.

FlowFerovTranslation

FlowFerovTranslation is reserved for internal use.

Field	Field Type	Description
`complexValues`	FlowComplexLiteralTranslation	Reserved for internal use.
`stringValues`	string	Reserved for internal use.

FlowComplexLiteralTranslation

FlowComplexLiteralTranslation is reserved for internal use.

Field	Field Type	Description
`customAspectKey`	string	Reserved for internal use.
`value`	string	Reserved for internal use.

FlowStageTranslation

FlowStageTranslation contains details for a translation of a stage in a flow version. For more details, see FlowStage in Flow.

Available in API version 43.0 and later.

Field	Field Type	Description
label	string	A translated label for the stage.
name	string	Required. An API name for the stage.

FlowTextTemplateTranslation

FlowTextTemplateTranslation is available only in flows created via Salesforce Surveys and represents the translation details for the text on all the pages of a survey.

Available in API version 45.0 and later.

Field	Field Type	Description
name	string	Required. Unique name for the text template.
text	string	Translated text for the text template.

IdentityVerificationFieldTranslation

Translates the UI components associated with identity verification fields.

Available in API version 54.0 and later.

Field	Field Type	Description
customFieldLabel	string	The custom label for the field that contains the verification data.
description	string	The identity verification field description.
label	string	A user-friendly name for IdentityVerificationFieldTranslation.
name	string	Required. The name of the identity verification field.

Declarative Metadata Sample Definition

This is an example of an IdentityVerificationFieldTranslation component.

```
<?xml version="1.0" encoding="UTF-8"?>
<Translations
    xmlns="http://soap.sforce.com/2006/04/metadata">
    <identityVerificationCustomFieldLabels>
        <description>Telefono Numero</description>
        <label>Telefono Numero</label>
        <name>Sample93Phone</name>
    </identityVerificationCustomFieldLabels>
```

```
    <identityVerificationCustomFieldLabels>
        <description>Nombre de la Cuenta</description>
        <label>Nombre de la Cuenta</label>
        <name>Sample93Account</name>
    </identityVerificationCustomFieldLabels>
    <identityVerificationCustomFieldLabels>
        <name>Sample93PostalCode</name>
    </identityVerificationCustomFieldLabels>
    <identityVerificationCustomFieldLabels>
        <name>Sample93AccountName</name>
        <description>Nombre</description>
        <label>Nombre</label>
    </identityVerificationCustomFieldLabels>
</Translations>
```

GlobalPicklistTranslation

Note: GlobalPicklistTranslation is available in API version 37.0 only and is removed from later versions.

GlobalPicklistTranslation contains details for a global picklist translation.

Translations are stored in a file with a format of *globalPicklistName__e-lang*.objectTranslation, where *globalPicklistName__e* is the global picklist name and *lang* is the translation language. To reference a global picklist translation value, use *globalPicklistName__e.value1*, where *value1* is the translated value for the user interface.

Here's what translations look like for a global picklist.

```
<?xml version="1.0" encoding="UTF-8"?>
<Translations xmlns="http://soap.sforce.com/2006/04/metadata">
<globalPicklists>
    <name>transpicklist</name>
    <picklistValues>
        <masterLabel>Three</masterLabel>
        <translation>Trois</translation>
    </picklistValues>
    <picklistValues>
        <masterLabel>Four</masterLabel>
        <translation>Quatre</translation>
    </picklistValues>
</globalPicklists>
</Translations>
```

Field	Field Type	Description
name	string	Required. Represents the name of a global picklist to be translated.
picklistValues	PicklistValueTranslation[]	A list of picklist values from global picklists to be translated.

GlobalQuickActionTranslation

GlobalQuickActionTranslation contains details for the global translation of a quick action. For more information, see QuickAction.

Field	Field Type	Description
`aspect`	string	Identifies which quick action label the translated text belongs to. Use this field only when you want to use different strings for the quick action's field label and informational message. Valid values are `Master` and `InfoMessage`. Available in API version 53.0 and later.
`label`	string	Required. The translated quick action name, globally.
`name`	string	Required. The quick action name.

PipelineInspMetricConfigTranslation

PipelineInspMetricConfigTranslation contains details for the translation of Pipeline Inspection forecast category metric settings. Available in API version 57.0 and later.

Field	Field Type	Description
`label`	string	Required. The translated Pipeline Inspection metric configuration name.
`name`	string	Required. The name of the Pipeline Inspection metric configuration.

PromptTranslation

PromptTranslation contains metadata for the translation of a prompt, which is part of In-App Guidance. Available in API Version 48.0 and later.

Field	Field Type	Description
`description`	string	The prompt description.
`label`	string	The translated prompt name.
`name`	string	Required. The name of the prompt.
`promptVersions`	PromptVersionTranslation	A list of the prompt version translations.

PromptVersionTranslation

PromptVersionTranslation contains details for translation of a prompt, which is part of In-App Guidance. Available in API Version 48.0 and later.

Field	Field Type	Description
`actionButtonLabel`	string	The label for the prompt's action button.
`actionButtonLink`	string	The URL for the prompt's action button.

Field	Field Type	Description
body	string	The body text of the prompt.
description	string	The prompt description.
dismissButtonLabel	string	The label for the floating prompt's dismiss button.
header	string	The header for the docked prompt.
imageAltText	string	The alt text for a prompt's image. Available in API version 53.0 and later.
imageLink	string	The URL for a prompt's image. Available in API version 53.0 and later.
label	string	The translated prompt name.
name	string	Required. The name of the prompt.
title	string	The title of the prompt.
videoLink	string	The URL for the docked prompt's video.

ReportTypeTranslation

ReportTypeTranslation contains details for a translation of a custom report type. For more details, see ReportType.

Field	Field Type	Description
description	string	The translated report type description.
label	string	The translated report type name.
name	string	Required. The name of the report type.
sections	ReportTypeSectionTranslation[]	A list of report type section translations.

ReportTypeSectionTranslation

ReportTypeSectionTranslation contains details for a report type section translation.

Field	Field Type	Description
columns	ReportTypeColumnTranslation[]	A list of report type column translations.
label	string	The translated report type section name.
name	string	Required. The name of the report type section.

ReportTypeColumnTranslation

ReportTypeColumnTranslation contains details for a report type column translation.

Field	Field Type	Description
label	string	Required. The translated report type column name.
name	string	Required. The report type column name.

ScontrolTranslation

Important: Visualforce pages supersede s-controls. Organizations that haven't previously used s-controls can't create them. Existing s-controls are unaffected and can still be edited.

ScontrolTranslation contains details for a translation of an s-control. For more information, see "About S-Controls" in Salesforce Help.

Field	Field Type	Description
label	string	Required. The translated s-control name.
name	string	Required. The name of the s-control.

Declarative Metadata Sample Definition

This sample XML definition shows a translations component.

```
<?xml version="1.0" encoding="UTF-8"?>
<Translations xmlns="http://soap.sforce.com/2006/04/metadata">
    <customApplications>
    <label>Angebot-Manager</label>
        <name>Quote Manager</name>
    </customApplications>
    <customLabels>
    <label>Dieses ist ein manuelles Angebot</label>
        <name>quoteManual</name>
    </customLabels>
</Translations>
```

Usage

When you use the `retrieve()` call to get translations, the files returned in the `.translations` folder only include translations for the other metadata types referenced in `package.xml`. For example, this `package.xml` file contains `types` elements that match all custom applications, custom labels, web links defined in home page components, custom tabs, report types, and s-controls. Translations for all these metadata types are returned because each metadata type is explicitly listed in `package.xml`.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
       <members>*</members>
       <name>CustomApplication</name>
    </types>
    <types>
       <members>*</members>
       <name>CustomLabels</name>
```

```
    </types>
    <types>
        <members>*</members>
        <name>CustomPageWebLink</name>
    </types>
    <types>
        <members>*</members>
        <name>CustomTab</name>
    </types>
    <types>
        <members>*</members>
        <name>ReportType</name>
    </types>
    <types>
        <members>*</members>
        <name>Scontrol</name>
    </types>
    <types>
        <members>*</members>
         <name>Translations</name>
    </types>
    <version>64.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SEE ALSO:

CustomLabels

UiFormatSpecificationSet

Represents a set of rules that define the style and visibility of conditional field formatting on Dynamic Forms-enabled Lightning page field instances.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

Note: A UiFormatSpecificationSet is referred to as a conditional formatting ruleset in the rest of the Salesforce documentation and UI.

File Suffix and Directory Location

UiFormatSpecificationSet components have the suffix `.uiFormatSpecificationSet` and are stored in the `uiFormatSpecificationSets` folder.

Version

UiFormatSpecificationSet components are available in API version 62.0 and later.

Special Access Rules

There are no additional access requirements that are specific to this type.

Fields

Field Name	Description
`field`	**Field Type** string **Description** Required. The object field that the conditional formatting is associated with.
`formatType`	**Field Type** FormatType (enumeration of type string) **Description** Required. The type of conditional formatting associated with the field. Values are: • `ICON`
`masterLabel`	**Field Type** string **Description** Required. The label for the conditional formatting ruleset, which displays in Setup.
`sobjectType`	**Field Type** string **Description** Required. The object the ruleset is associated with.
`uiFormatSpecifications`	**Field Type** UiFormatSpecification[] **Description** The list of rules contained in the ruleset.

UiFormatSpecification

A single rule in the ruleset.

Field Name	Description
`formatProperties`	**Field Type** string **Description** Required. The properties for a given `formatType` in JSON format.
`formatType`	**Field Type** FormatType (enumeration of type string) **Description** Required. The type of conditional formatting associated with the field when the rule evaluates to `true`. Values are: • `ICON`
`order`	**Field Type** int **Description** Required. A numerical value representing the conditional formatting rule's position in the evaluation order.
`visibilityRule`	**Field Type** UiFormulaRule **Description** A set of one or more filters that define the conditions under which the conditional formatting appears on the field. If the visibility rule evaluates to `true`, the formatting displays on the field. If `false`, it doesn't display. If this field is `null`, the formatting displays by default.

UiFormulaRule

A set of one or more filters that define the conditions under which conditional field formatting displays on a Dynamic Forms-enabled Lightning page field instance. For example, you could construct a filter that causes conditional formatting to display only when the Amount field is greater than $1,000,000.

Field Name	Description
`booleanFilter`	**Field Type** string

Field Name	Description
	Description Specifies advanced filter conditions such as `1 AND 2`.
`criteria`	**Field Type** UiFormulaCriterion[] **Description** List of one or more filters that, when evaluated, determine conditional field formatting visibility.

UiFormulaCriterion

A single filter that when evaluated, helps define conditional formatting visibility on a Dynamic Forms-enabled Lightning page field instance.

Field Name	Description
`leftValue`	**Field Type** string **Description** Required. The field upon which the filter is based. For example, `AMOUNT`.
`operator`	**Field Type** string **Description** Required. Defines the operator used to filter the data. Valid values are: • `CONTAINS` • `EQUAL` • `NE`—not equal • `GT`—greater than • `GE`—greater than or equal • `LE`—less than or equal • `LT`—less than
`rightValue`	**Field Type** string **Description** The value by which you want to evaluate the formatting visibility. For example, `1000000`.

Declarative Metadata Sample Definition

The following is an example of an UiFormatSpecificationSet component.

```
<?xml version="1.0" encoding="UTF-8"?>
<UiFormatSpecificationSet xmlns="http://soap.sforce.com/2006/04/metadata">
    <field>Contact.Customer_Sentiment__c</field>
    <formatType>ICON</formatType>
    <masterLabel>Sentiment Score</masterLabel>
    <sobjectType>Contact</sobjectType>
    <uiFormatSpecifications>
        <formatProperties>{&quot;icon&quot;:&quot;happy_face&quot;,
&quot;iconColor&quot;:&quot;green&quot;}</formatProperties>
        <formatType>ICON</formatType>
        <order>1</order>
        <visibilityRule>
            <criteria>
                <leftValue>{!Record.Customer_Sentiment__c}</leftValue>
                <operator>EQUAL</operator>
                <rightValue>Happy</rightValue>
            </criteria>
        </visibilityRule>
    </uiFormatSpecifications>
    <uiFormatSpecifications>
        <formatProperties>{&quot;icon&quot;:&quot;neutral_face&quot;,
&quot;iconColor&quot;:&quot;gray&quot;}</formatProperties>
        <formatType>ICON</formatType>
        <order>2</order>
        <visibilityRule>
            <criteria>
                <leftValue>{!Record.Customer_Sentiment__c}</leftValue>
                <operator>EQUAL</operator>
                <rightValue>Neutral</rightValue>
            </criteria>
        </visibilityRule>
    </uiFormatSpecifications>
    <uiFormatSpecifications>
        <formatProperties>{&quot;icon&quot;:&quot;sad_face&quot;,
&quot;iconColor&quot;:&quot;red&quot;}</formatProperties>
        <formatType>ICON</formatType>
        <order>3</order>
        <visibilityRule>
            <criteria>
                <leftValue>{!Record.Customer_Sentiment__c}</leftValue>
                <operator>EQUAL</operator>
                <rightValue>Unhappy</rightValue>
            </criteria>
        </visibilityRule>
    </uiFormatSpecifications>
</UiFormatSpecificationSet>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
```

```
    <types>
        <members>Sentiment_Score</members>
        <name>UiFormatSpecificationSet</name>
    </types>
    <version>62.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

UIObjectRelationConfig

Represents the admin-created configuration of the object relation UI component.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

UIObjectRelationConfig components have the suffix `.uiObjectRelationConfig` and are stored in the `uiObjectRelationConfigs` folder.

Version

UIObjectRelationConfig components are available in API version 54.0 and later.

Special Access Rules

You must be a Health Cloud or Life Sciences Cloud customer to use this metadata type

Fields

Field Name	Description
`contextObject`	**Field Type** string **Description** Required. The object that provides the context for this object relation configuration.

Field Name	Description
`contextObjectRecordType`	**Field Type** string **Description** The record type of the context object for this configuration, if applicable.
`directRelationshipField`	**Field Type** string **Description** For direct relationships, the child relationship field on the related object that matches the context object.
`indirectObjectContextField`	**Field Type** string **Description** For indirect relationships, the field on the junction object that matches the context object.
`indirectObjectRelatedField`	**Field Type** string **Description** For indirect relationships, the field on the junction object that matches the related object.
`indirectRelationshipObject`	**Field Type** string **Description** For indirect relationships, the junction object representing the relationship between the related object and its context object.
`isActive`	**Field Type** boolean **Description** Indicates whether the configuration is active (`true`) or not (`false`).
`masterLabel`	**Field Type** string **Description** Required. Label for the UIObjectRelationConfig. In the UI, this field is UI Object Relation Config.
`relatedObject`	**Field Type** string

Field Name	Description
	Description Required. The object containing the data that this object relation configuration displays.
`relatedObjectRecordType`	**Field Type** string **Description** The record type of the related object for this configuration.
`relationshipType`	**Field Type** ObjectRelationshipType (enumeration of type string) **Description** Required. A string indicating the type of relationship between the related object and context object. Valid values are: • `Direct` • `Indirect` • `InverseDirect` • `Self`
`UIObjectRelationFieldConfigs`	**Field Type** UIObjectRelationFieldConfig[] **Description** Provides a configuration for an object relation field on a specific row of content.

UIObjectRelationFieldConfig

Represents a configuration for a single row of content on a specific object relation configuration.

Field Name	Description
`displayLabel`	**Field Type** string **Description** Required. A string containing the user-defined label for this field, to be displayed on each object relation of this type.

Field Name	Description
queryText	**Field Type** string **Description** Required. A case-insensitive template query for generating the content in this field.
rowOrder	**Field Type** int **Description** Required. Determines the top-to-bottom display order of this field on the object relation UI.

Declarative Metadata Sample Definition

This is an example of a UIObjectRelationConfig component.

```
<?xml version="1.0" encoding="UTF-8"?>
<UIObjectRelationConfig xmlns="http://soap.sforce.com/2006/04/metadata">
    <UIObjectRelationFieldConfigs>
        <displayLabel>Address:</displayLabel>
        <queryText>{
    "startNode": {
        "initialObject": "RelatedObject"
    },
    "traversalNodes": [],
    "fieldNode": {
        "fieldEnumOrId": "ShippingAddress"
    }
}</queryText>
        <rowOrder>1</rowOrder>
    </UIObjectRelationFieldConfigs>
    <UIObjectRelationFieldConfigs>
        <displayLabel>Phone:</displayLabel>
        <queryText>{
    "startNode": {
        "initialObject": "RelatedObject"
    },
    "traversalNodes": [],
    "fieldNode": {
        "fieldEnumOrId": "Phone"
    }
}</queryText>
        <rowOrder>2</rowOrder>
    </UIObjectRelationFieldConfigs>
    <UIObjectRelationFieldConfigs>
        <displayLabel>Fax:</displayLabel>
        <queryText>{
```

```
    "startNode": {
        "initialObject": "RelatedObject"
    },
    "traversalNodes": [],
    "fieldNode": {
        "fieldEnumOrId": "Fax"
    }
}</queryText>
        <rowOrder>3</rowOrder>
    </UIObjectRelationFieldConfigs>
    <UIObjectRelationFieldConfigs>
        <displayLabel>Parent Organization:</displayLabel>
        <queryText>{
    "startNode": {
        "initialObject": "RelatedObject"
    },
    "traversalNodes": [
        {
            "destinationObjectEnumOrId": "Account",
            "fieldEnumOrId": "ParentId",
            "traversalDirection": "parent"
        }
    ],
    "fieldNode": {
        "fieldEnumOrId": "Name"
    }
}</queryText>
        <rowOrder>4</rowOrder>
    </UIObjectRelationFieldConfigs>
    <contextObject>Contact</contextObject>
    <directRelationshipField>AccountId</directRelationshipField>
    <isActive>true</isActive>
    <masterLabel>Sample Primary Account Configuration</masterLabel>
    <relatedObject>Account</relatedObject>
    <relationshipType>Direct</relationshipType>
    <indirectObjectRelatedField></indirectObjectRelatedField>
    <indirectObjectContextField></indirectObjectContextField>
    <contextObjectRecordType></contextObjectRecordType>
    <indirectRelationshipObject></indirectRelationshipObject>
    <relatedObjectRecordType></relatedObjectRecordType>
</UIObjectRelationConfig>
```

This is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
  <types>
      <members>*</members>
      <name>UIObjectRelationConfig</name>
  </types>
    <version>54.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

UserAccessPolicy

Represents a user access policy.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

UserAccessPolicy components have the suffix `.useraccesspolicy` and are stored in the `useraccesspolicies` folder.

Version

UserAccessPolicy components are available in API version 57.0 and later.

Special Access Rules

To create or modify user access policies, users must have the Manage User Access Policies permission.

Fields

Field Name	Description
`booleanFilter`	**Field Type** string **Description** Required. The logic that determines how your user criteria filters are applied in the user access policy. For example, if you have two user access policy filters with the `sortOrder` equal to `1` and `2`, respectively, the `booleanFilter` can be `1 AND 2` or `1 OR 2`.
`description`	**Field Type** string **Description** Description of the user access policy.

Field Name	Description
`isProtected`	**Field Type** boolean **Description** An auto-generated value that doesn't impact the behavior of the metadata type. The default value is `false`.
`masterLabel`	**Field Type** string **Description** Required. A user-friendly name for the user access policy, which is defined when the user access policy is created.
`order`	**Field Type** int **Description** Indicates the order for which active policy is applied when a user meets the criteria for multiple policies. Must be an integer from 0 to 10,000. Only the active policy with the lowest `order` value is applied. This field is required only if the `status` field is set to `Active`. Available in API version 61.0 and later.
`status`	**Field Type** UserAccessPolicyStatus (enumeration of type string) **Description** Required. The status of the user access policy. Values are: • `Active` • `Completed` • `Design` • `Failed` • `Migrate` • `Testing` • `Updating` If you deploy a policy with a status of `Active`, the status is changed to `Design`. A Salesforce admin can then set the status to `Active` by automating the policy in Setup.
`triggerType`	**Field Type** UserAccessPolicyTriggerType (enumeration of type string)

Field Name	Description
	Description The type of user record trigger for which this user access policy runs. Values are: • `Create`—The user access policy runs when a user who matches the policy criteria is created. • `CreateAndUpdate`—The user access policy runs when a user who matches the policy criteria is either created or updated. • `Update`—The user access policy runs when a user who matches the policy criteria is updated.
`userAccessPolicyActions`	**Field Type** UserAccessPolicyAction[] **Description** The actions applied by the user access policy to grant access to or revoke access from an access mechanism.
`userAccessPolicyFilters`	**Field Type** UserAccessPolicyFilter[] **Description** The filters used to define the users that the user access policy is applied to.

UserAccessPolicyAction

Represents an action applied by a user access policy.

Field Name	Description
`action`	**Field Type** UserAccessPolicyActionType (enumeration of type string) **Description** Required. Indicates whether the user access policy grants or revokes the target access mechanism. Values are: • `Grant` • `Revoke`
`target`	**Field Type** string **Description** Required. Developer name of the access mechanism that the user access policy applies.

Field Name	Description
`type`	**Field Type** UserAccessPolicyActionTargetType (enumeration of type string) **Description** Required. The type of access mechanism that the user access policy applies. Values are: • `Group` • `PackageLicense` • `PermissionSet` • `PermissionSetGroup` • `PermissionSetLicense` • `Queue`

UserAccessPolicyFilter

Represents a user criteria filter for a user access policy.

Field Name	Description
`columnName`	**Field Type** string **Description** If `type` is set to `User`, this is the user field that your user criteria filter is based on. If you set `type` to any value other than `User`, then this field isn't used.
`operation`	**Field Type** UserAccessPolicyFilterOperation (enumeration of type string) **Description** Required. The operator of the user criteria filter. Values are: • `equals` • `equalsIgnoreCase`—Available in API version 59.0 and later. • `in`— Available in API version 58.0 and later. • `includes`—Available in API version 59.0 and later. • `notEquals` Select `in` if you want to reference multiple profiles or roles in the same user criteria filter via the `target` field.
`sortOrder`	**Field Type** int

Field Name	Description
	Description Required. The numeric reference used to identify the specific user criteria filter.
`target`	**Field Type** string **Description** Required. If `type` is set to `User`, then set this field to `User` as well. If `type` is set to any other value, then set this field to the developer name of the specific resource used in the user criteria filter.
`type`	**Field Type** UserAccessPolicyFilterTargetType (enumeration of type string) **Description** Required. The type of resource that the user criteria filter is based on. Values are: • `Group` • `PackageLicense` • `PermissionSet` • `PermissionSetGroup` • `PermissionSetLicense` • `Profile` • `Queue` • `User` • `UserRole`
`value`	**Field Type** string **Description** If `type` is set to `User`, this field is the value of the user field specified in `columnName` that your user filter is operating on. If you set `type` to any value other than `User`, then this field isn't used.

Declarative Metadata Sample Definition

The following is an example of a UserAccessPolicy component.

```
<?xml version="1.0" encoding="UTF-8"?>
<UserAccessPolicy xmlns="http://soap.sforce.com/2006/04/metadata">
    <booleanFilter>1 AND 2</booleanFilter>
    <description>Policy to assign Sales Rep PSG to active Sales Reps.</description>
    <masterLabel>Sales Rep Migration</masterLabel>
```

```
    <order>3</order>
    <status>Design</status>
    <triggerType>CreateAndUpdate</triggerType>
    <userAccessPolicyActions>
        <action>Grant</action>
        <target>SalesRepPSG</target>
        <type>PermissionSetGroup</type>
    </userAccessPolicyActions>
    <userAccessPolicyFilters>
        <operation>equals</operation>
        <sortOrder>1</sortOrder>
        <target>SalesRepCustomProfile</target>
        <type>Profile</type>
    </userAccessPolicyFilters>
    <userAccessPolicyFilters>
        <columnName>IsActive</columnName>
        <operation>equals</operation>
        <sortOrder>2</sortOrder>
        <target>User</target>
        <type>User</type>
        <value>true</value>
    </userAccessPolicyFilters>
</UserAccessPolicy>
```

To reference multiple profiles or roles, in UserAccessPolicyFilter, set the `operator` field to `in`. Then, reference the resources' developer names in the `target` field, separated by commas.

```
<?xml version="1.0" encoding="UTF-8"?>
<UserAccessPolicy xmlns="http://soap.sforce.com/2006/04/metadata">
    <booleanFilter>1</booleanFilter>
    <description>Policy to remove AMER Sales group from employees with one of two
roles</description>
    <masterLabel>Remove AMER Sales Group</masterLabel>
    <status>Design</status>
    <userAccessPolicyActions>
        <action>Revoke</action>
        <target>AMERSalesPublicGroup</target>
        <type>Group</type>
    </userAccessPolicyActions>
    <userAccessPolicyFilters>
        <operation>in</operation>
        <sortOrder>1</sortOrder>
        <target>SalesOps,InsideSalesRep</target>
        <type>UserRole</type>
    </userAccessPolicyFilters>
</UserAccessPolicy>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>UserAccessPolicy</name>
    </types>
```

```
    <version>61.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

UserAuthCertificate

Represents a PEM-encoded user certificate. These certificates are associated with a user, and externally uploaded. The uploaded certificate is used to authenticate the user.

This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

UserAuthCertificate components have the suffix `.userAuthCertificate` and are stored in the `userAuthCertificates` folder.

Version

UserAuthCertificate components are available in API version 50.0 and later.

Fields

Field Name	Field Type	Description
`developerName`	string	Required: The name of the certificate with an underscore between words. **Note:** Only users with View DeveloperName OR View Setup and Configuration permission can view, group, sort, and filter this field.
`expirationDate`	dateTime	Required. The date on which the certificate expires.
`masterLabel`	string	Required. A user-friendly name that you create for the certificate. Limited to 64 characters.
`serialNumber`	string	Required. The serial number for the certificate.
`user`	string	Required: The user's name.

Declarative Metadata Sample Definition

The following is an example of a UserAuthCertificate component.

```
<UserAuthCertificate xmlns="http://soap.sforce.com/2006/04/metadata"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <content xsi:nil="true"/>
    <developerName>ND_user_cert</developerName>
    <expirationDate>2030-10-01T08:30:00.000Z</expirationDate>
    <masterLabel>ND user cert</masterLabel>
    <serialNumber>1401</serialNumber>
    <user>005RM000001Zn1E</user>
</UserAuthCertificate>
```

The following is an example `package.xml` that references the previous definition.

```
Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>UserAuthCertificate</name>
    </types>
    <version>50.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

UserCriteria

Represents the member criteria to use in Experience Cloud site moderation rules. This type extends the Metadata metadata type and inherits its `fullName` field..

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

UserCriteria components have the suffix `site_name.user_criteria_developer_name.userCriteria` and are stored in the `UserCriteria` folder.

Version

UserCriteria components are available in API version 39.0 and later.

Special Access Rules

To view, create, edit, and delete moderation rules, you need the Manage Experiences or Create and Set Up Experiences permission. As of Spring '20 and later, only users with permission to edit moderation rules can access this object.

Fields

Field Name	Field Type	Description
`creationAgeInSeconds`	int	If specified, includes only users that were created within a specific time frame.
`description`	string	The description of the user criteria.
`lastChatterActivityAgeInSeconds`	int	If specified, includes only members that have posted or commented in the Experience Cloud site within a specific time frame.
`masterLabel`	string	Name of the user criteria.
`userTypes`	NetworkUserType enumeration (of type string)	The member type to use in moderation rules. Valid values are: • Internal • Customer • Partner

Declarative Metadata Sample Definition

The following is an example of a UserCriteria component.

```
<?xml version="1.0" encoding="UTF-8"?>
<UserCriteria xmlns="http://soap.sforce.com/2006/04/metadata">
    <masterLabel>Customer and Partner Members</masterLabel>
    <description>Member criteria matches customer and partner member</description>
    <userTypes>Partner</userTypes>
    <userTypes>Customer</userTypes>
</UserCriteria>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

UserProfileSearchScope

Reserved for internal use.

UserProvisioningConfig

Represents information to use during a user provisioning request flow, such as the attributes for an update. This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

UserProvisioningConfig components have the suffix `.userProvisioningConfig` and are stored in the `UserProvisioningConfigs` directory.

Version

UserProvisioningConfig components are available in API version 49.0 and later.

Fields

Field Name	Field Type	Description
`approvalRequired`	string	Indicates whether approvals are required for provisioning users for the associated connected app. If the value is null, no approval is required.
`connectedApp`	string	The ID of the connected app for which users are being provisioned.
`enabled`	boolean	Indicates whether user provisioning is enabled for the associated connected app (`true`) or not (`false`). Default setting is `false`.
`enabledOperations`	string	Lists the operations, as comma-separated values, that create a user provisioning request for the associated connected app. Allowed values are: • `Create` • `Update` • `EnableAndDisable` (activation and deactivation) • `SuspendAndRestore` (freeze and unfreeze)
`flow`	string	User Provisioning flow type which includes a reference to the Apex `UserProvisioningPlugin` class. The flow calls the third-party service's API to manage user account provisioning on that system.
`masterLabel`	string	The primary label for this object. This value is the internal label that doesn't get translated.
`namedCredential`	string	The Salesforce ID of the named credential that's used for a request. The named credential identifies the third-party system and the third-party authentication settings.
`notes`	string	Serves as a place for admins to add any additional information about the configuration. This field is for internal reference only, and is not used by any process.
`onUpdateAttributes`	string	Lists the user attributes, as comma-separated values, that generate a user provisioning request during an update.
`reconFilter`	string	When collecting and analyzing users on a third-party system, the plug-in uses this filter to limit the scope of the collection.

Field Name	Field Type	Description
`userAccountMapping`	string	Stores the attributes used to link the Salesforce user to the account on the third-party system, in JSON format. For example: `{"linkingSalesforceUserAttribute":"Username", "linkingTargetUserAttribute":"Email"}`

Declarative Metadata Sample Definition

The following is an example of a UserProvisioningConfig component.

```
<?xml version="1.0" encoding="UTF-8"?>
<UserProvisioningConfig xmlns="http://soap.sforce.com/2006/04/metadata">
    <approvalRequired>True</approvalRequired>
    <enabled>true</enabled>
    <enabledOperations>NA</enabledOperations>
    <connectedApp>ExampleApp</connectedApp>
    <masterLabel>label</masterLabel>
    <notes>note</notes>
    <onUpdateAttributes>attri</onUpdateAttributes>
    <reconFilter>filter</reconFilter>
    <userAccountMapping>mapping</userAccountMapping>
</UserProvisioningConfig>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>UPC</members>
        <name>UserProvisioningConfig</name>
    </types>
    <version>49.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

VirtualVisitConfig

Represents an external video provider configuration, which relays events from Salesforce to the provider.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

VirtualVisitConfig components have the suffix `.virtualVisitConfig` and are stored in the `VirtualVisitConfigs` folder.

Version

VirtualVisitConfig components are available in API version 54.0 and later.

Special Access Rules

Access to this metadata type requires at least one of these preferences:

- Video Calls: Org Pref (VideoVisits) Org preference
- Industries Einstein: Intelligent Form Reader (EinsteinDocReader)
- Industries Einstein: Sentiment Insights Account (IESentimentAnalysis)
- Natural Language Processing: Key phrase extraction and entity detection (NLPServiceEnabled) Org Preference and the NLP: Key phrase extraction (KeyPhrasePrefEnabled) Org Preference
- Natural Language Processing (NLPServicePrefEnabled) Org Preference

Fields

Field Name	Description
`comprehendServiceType`	**Field Type** VirtualVisitComprehendServiceType (enumeration of type string) **Description** Specifies the type of service used to convert speech into text or to analyze the converted speech text. Valid values are: • `ComprehendMedicalService` • `ComprehendService`
`developerName`	**Type** string **Properties** Filter, Group, Sort **Description** The unique name of the object in the API. This name can contain only underscores and alphanumeric characters, and must be unique in your org. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. In managed packages, this field prevents naming conflicts on package

Field Name	Description
	installations. With this field, a developer can change the object's name in a managed package and the changes are reflected in a subscriber's organization. Label is Record Type Name. This field is automatically generated, but you can supply your own value if you create the record using the API.
`experienceCloudSiteUrl`	**Field Type** string **Description** The URL of the Digital Experience site where the Video Call component is available to portal or guest users.
`externalMsgServiceIdentifier`	**Field Type** string **Description** For internal use only.
`externalRoleIdentifier`	**Field Type** string **Description** The ID of the role that's used to allow users to join a video call and to grant them temporary access to certain functions needed to participate in the call.
`externalUserIdentifier`	**Field Type** string **Description** For internal use only.
`fullName`	**Type** string **Properties** Create, Group, Nillable **Description** The full name of the VirtualVisitConfig type in Metadata API. The full name can include a namespace prefix. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.
`isProtected`	**Field Type** boolean **Description** An auto-generated value that doesn't currently impact the behavior of the metadata type.

Field Name	Description
masterLabel	**Field Type** string **Description** Required. A user-friendly name for VirtualVisitConfig, which is defined when the VirtualVisitConfig is created.
messagingRegion	**Field Type** string **Description** The region where the waiting room and messaging channel data is processed and stored. Available in API version 57.0 and later.
namedCredential	**Field Type** string **Description** The named credential record used to authenticate and authorize a video call vendor's account.
storageBucketName	**Field Type** string **Description** The name of the storage bucket that stores the meeting transcript.
usageType	**Field Type** VirtualVisitUsageType (enumeration of type string) **Description** The name of the Salesforce feature for which the video call configuration record is created. Valid values are: • CHIME • ENTITY_DETECTION • INTELLIGENT_FORM_READER • KEY_PHRASE_EXTRACTION • SENTIMENT_ANALYSIS
videoCallApptTypeValue	**Field Type** string **Description** The default Appointment Type picklist values from the Service Appointment object that represent a video appointment type. Use semicolons to separate multiple values.

Field Name	Description
videoControlRegion	**Field Type** string **Description** The region where API calls related to Video Calls are made. Available in API version 57.0 and later.
visitRegion	**Field Type** VirtualVisitVisitRegion (enumeration of type string) **Description** The region where the Video Call audio and video data is processed. Valid values are: • af-south-1 • ap-east-1 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-south-1 • ap-southeast-1 • ap-southeast-2 • ca-central-1 • eu-central-1 • eu-north-1 • eu-south-1 • eu-west-1 • eu-west-2 • eu-west-3 • me-south-1 • sa-east-1 • us-east-1 • us-east-2 • us-west-1 • us-west-2

Declarative Metadata Sample Definition

This is an example of a VirtualVisitConfig component.

```
<?xml version="1.0" encoding="UTF-8"?>
 <VirtualVisitConfig xmlns="http://soap.sforce.com/2006/04/metadata">
     <usageType>CHIME</usageType>
```

```
    <visitRegion>us-east-1</visitRegion>
    <masterLabel>vvconfig1</masterLabel>
    <experienceCloudSiteUrl>videocall_c@testcloudurl.com</experienceCloudSiteUrl>
    <namedCredential>SampleNamedCredential</namedCredential>
    <comprehendServiceType>ComprehendService</comprehendServiceType>
    <storageBucketName>comprehendbucket</storageBucketName>
    <isProtected>false</isProtected>
</VirtualVisitConfig>
```

This is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <namespacePrefix>[namespacePrefix]</namespacePrefix>
    <fullName>deployPackage</fullName>
    <types>
        <members>*</members>
        <name>VirtualVisitConfig</name>
    </types>
    <types>
        <members>*</members>
        <name>NamedCredential</name>
    </types>
    <version>55.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

WaveAnalyticAssetCollection

Represents a collection of Analytics assets. This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

WaveAnalyticsAssetCollection components have the suffix `.collection` and are stored in the `wave` folder.

Version

WaveAnalyticsAssetCollection components are available in API version 58.0 and later.

Fields

Field Name	Field Type	Description
collectionType	string	The collection type.
color	string	The display color for the collection.
description	string	The description that appears in the user interface.
folder	string	The internal API name of the folder or application.
items	WaveAnalyticAssetCollectionItem	A list of Analytics asset items.
label	string	The label for the collection.
shares	FolderShare	The folder sharing rules.

Declarative Metadata Sample Definition

The following is an example of a WaveAnalyticsAssetCollection component.

```
<?xml version="1.0" encoding="UTF-8"?>
<WaveAnalyticsAssetCollection xmlns="http://soap.sforce.com/2006/04/metadata">
  <collectionType>static</collectionType>
  <color>#1b96ff</color>
  <description>A collection of my Dashboards</description>
  <folder>Shared</folder>
  <label>My Dashboard Collection</label>
  <items>
     <item>
        <asset>Dashboard One</asset>
        <assetType>dashboard</assetType>
        <sortOrder>1</sortOrder>
     </item>
     <item>
        <asset>Dashboard Two</asset>
        <assetType>dashboard</assetType>
        <sortOrder>2</sortOrder>
     </item>
  </items>
  <shares>
    <accessLevel>EditAllContents</accessLevel>
    <sharedTo>shareswith@org.ee</sharedTo>
    <sharedToType>User</sharedToType>
  </shares>
</WaveAnalyticsAssetCollection>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character * (asterisk) in the package.xml manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

WaveAnalyticAssestCollectionItem

WaveAnalyticAssestCollectionItem represents an Analytics asset item.

Field	Field Type	Description
`asset`	string	The asset name.
`assetType`	string	The asset type.
`sortOrder`	int	The sort order for the asset.

WaveApplication

Represents the Analytics application. This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

WaveApplication components have the suffix `.wapp` and are stored in the `wave` folder.

Version

WaveApplication components are available in API version 37.0 and later.

Fields

Field Name	Field Type	Description
`assetIcon`	string	The icon that represents the Analytics application.
`description`	string	The description that appears in the user interface.
`folder`	string	The internal api name of the folder or application.
`masterLabel`	string	The user interface label name of the folder or application.
`shares`	FolderShare	The folder sharing rules.
`templateOrigin`	string	The internal (unique) name of the template used to create the application. This field is blank if the application wasn't created from a template.
`templateVersion`	string	The version assigned to the application template by the template's creator. This field is blank if the application wasn't created from a template.

Declarative Metadata Sample Definition

The following is an example of a WaveApplication component.

```
<?xml version="1.0" encoding="UTF-8"?>
<WaveApplication xmlns="http://soap.sforce.com/2006/04/metadata">
  <assetIcon>/analytics/wave/web/proto/images/app/icons/11.png</assetIcon>
  <description>Application that shows my sales</description>
  <folder>edit</folder>
  <masterLabel>Sales Application</masterLabel>
  <shares>
   <accessLevel>EditAllContents</accessLevel>
   <sharedTo>shareswith@org.ee</sharedTo>
   <sharedToType>User</sharedToType>
  </shares>
</WaveApplication>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

WaveComponent

Represents the WaveComponent object in the Analytics application. This type extends the MetadataWithContent metadata type and inherits its `content` and `fullName` fields.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

When using Metadata API to work with Analytics components, consider that:

- Modifications to the `.wcomp` component are unsupported.

File Suffix and Directory Location

WaveComponent components have the suffix `.wcomp` and are stored in the `wave` folder.

Version

WaveComponent components are available in API version 51.0 and later.

Fields

Field Name	Field Type	Description
`application`	string	Required. The internal name of the application.
`description`	string	The component description that appears in the user interface.
`masterLabel`	string	Required. The component name that appears in the user interface.

Field Name	Field Type	Description
templateAssetSourceName	string	Links the component to the template used to create it. Null for assets not created from a template.

Declarative Metadata Sample Definition

The following is an example of a WaveComponent component.

```
<?xml version="1.0" encoding="UTF-8"?>
<WaveComponent xmlns="http://soap.sforce.com/2006/04/metadata"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <content xsi:nil="true"/>
    <application>dev__app</application>
    <masterLabel>Component1</masterLabel>
    <description>Component description</description>
</WaveComponent>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character * (asterisk) in the package.xml manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

WaveDataflow

Represents the WaveDataflow object in the Analytics application. This type extends the MetadataWithContent metadata type and inherits its content and fullName fields.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

WaveDataflow components have the suffix .wdf and are stored in the wave folder.

Version

WaveDataflow components are available in API version 37.0 and later.

Fields

Field Name	Field Type	Description
application	string	The name of the Analytics application the dataflow is connected to. This field is available in API version 48.0 and later.

Field Name	Field Type	Description
dataflowType	string	The type of the dataflow. Supported types are User and Prepared. The default value is User This field is available in API version 41.0 and later.
description	string	The dataflow description that appears in the user interface.
masterLabel	string	Required. The dataflow name that appears in the user interface.

Declarative Metadata Sample Definition

The following is an example of a WaveDataflow component.

```
<?xml version="1.0" encoding="UTF-8"?>
<WaveDataflow xmlns="http://soap.sforce.com/2006/04/metadata"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">    <content xsi:nil="true"/>
    <description>flow1</description>
    <masterLabel>flow1</masterLabel>
</WaveDataflow>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character * (asterisk) in the package.xml manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

WaveDashboard

Represents the WaveDashboard object in the Analytics application. This type extends the MetadataWithContent metadata type and inherits its content and fullName fields.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

When using Metadata API to work with Analytics dashboards, consider that:

- Modifications to the .wdash component are unsupported.
- Modifying or removing conditional formatting from the source org or .wdash component doesn't cause issues while deploying.
- Removing steps from the .wdash component causes deployment to the destination org to fail because the source dashboard fails validation.

File Suffix and Directory Location

WaveDashboard components have the suffix .wdash and are stored in the wave folder.

Version

WaveDashboard components are available in API version 37.0 and later.

Fields

Field Name	Field Type	Description
`application`	string	Required. The internal name of the application.
`dateVersion`	integer	The date version for the dashboard. Only available in v55.0 and above.
`description`	string	The dashboard description that appears in the user interface.
`masterLabel`	string	Required. The dashboard name that appears in the user interface.
`templateAssetSourceName`	string	Links the dashboard to the template used to create it. Null for assets not created from a template.

Declarative Metadata Sample Definition

The following is an example of a WaveDashboard component.

```
<?xml version="1.0" encoding="UTF-8"?>
<WaveDashboard xmlns="http://soap.sforce.com/2006/04/metadata"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <content xsi:nil="true"/>
  <application>dev__app</application>
  <masterLabel>Dashboard1</masterLabel>
  <description>somedesc</description>
</WaveDashboard>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

WaveDataset

Represents the WaveDataset object in the Analytics application. This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

WaveDataset components have the suffix `.wds` and are stored in the `wave` folder.

Version

WaveDataset components are available in API version 37.0 and later.

Fields

Field Name	Field Type	Description
`application`	string	Required. The internal name of the application.
`description`	string	The dataset description that appears in the user interface.
`masterLabel`	string	Required. The user interface label name of the dataset.
`templateAssetSourceName`	string	Links the dataset to the template used to create it. Null for assets not created from a template.
`type`	string	The type of the dataset. Dataset types include `Default`, `Live`,`StagedData`, and `Trended`.

Declarative Metadata Sample Definition

The following is an example of a WaveDataset component.

```
<WaveDataset>
    <application>SharedApp</application>
    <description>description</description>
    <masterLabel>datasetLabel</masterLabel>
    <type>Default</type>
</WaveDataset>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

WaveLens

Represents the WaveLens object in the Analytics application.

This type extends to MetadataWithContent metadata type and inherits its `content` and `fullName` fields.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

WaveLens components have the suffix `.wlens` and are stored in the `wave` folder.

Version

WaveLens components are available in API version 37.0 and later.

Fields

Field Name	Field Type	Description
`application`	string	Required. The internal name of the application.
`datasets`	string	A reference to the dataset used to create this lens.
`dateVersion`	int	The date version used for this lens.
`description`	string	The dashboard description that appears in the user interface.
`masterLabel`	string	Required. The user interface label name of the dashboard.
`templateAssetSourceName`	string	Links the lens to the template used to create it. Null for assets not created from a template.
`visualizationType`	string	Required. The visualization type to be used for this lens. Valid values are: • `calheatmap`—Calendar heat map • `comparisontable`—Comparison table • `heatmap`—Heat map • `hbar`—Horizontal bar • `hbarhdot`—Horizontal dot plot • `matrix`—Matrix • `parallelcoords`—Parallel coordinates • `pie`—Donut • `pivottable`—Pivot table • `scatter`—Scatter plot • `stackhbar`—Stacked horizontal bar • `stackvbar`—Stacked vertical bar • `time`—Time line • `valuestable`—Values table • `vbar`—Vertical bar • `vdot`—Vertical dot plot

Declarative Metadata Sample Definition

The following is an example of a WaveLens component.

```
<?xml version="1.0" encoding="UTF-8"?>
<WaveLens xmlns="http://soap.sforce.com/2006/04/metadata"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <content xsi:nil="true"/>
  <application>dev__app</application>
  <datasets>dev__abc</datasets>
  <masterLabel>lens1</masterLabel>
  <description>lens in shared app</description>
```

```
  <visualizationType>hbar</visualizationType>
</WaveLens>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

WaveRecipe

Represents the WaveRecipe type in an Analytics application. A recipe is a saved set of steps to perform on a specific source dataset or connected data. This type extends the MetadataWithContent metadata type and inherits its `content` and `fullName` fields.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

WaveRecipe components have the suffix `.wdpr` and are stored in the `wave` folder.

Version

WaveRecipe components are available in API version 41.0 and later.

Fields

Field Name	Field Type	Description
`application`	string	The internal name of the application.
`dataflow`	string	Required. The dataflow ID for the Analytics recipe.
`format`	string	The format of the current recipe definition. Valid values are: • `R2` - recipes created with Data Prep • `R3` - recipes created with Data Prep (API version 49.0)
`masterLabel`	string	Required. The recipe name that appears in the user interface.
`securityPredicate`	string	A filter condition that defines row-level access to records in a recipe.
`targetDatasetAlias`	string	The name of the dataset the recipe saves data results into.
`templateAssetSourceName`	string	Links the recipe to the template used to create it. Null for assets not created from a template.

Declarative Metadata Sample Definition

The following is an example of a WaveRecipe component.

```
<?xml version="1.0" encoding="UTF-8"?>
<WaveRecipe xmlns="http://soap.sforce.com/2006/04/metadata"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">    <content xsi:nil="true"/>
    <dataflow>02KB0000000b5c7MAA</dataflow>
    <format>R3</format>
    <masterLabel>recipe1</masterLabel>
    <securityPredicate>'UserId' == "$User.Id"</securityPredicate>
    <targetDatasetAlias>Dataset One</targetDatasetAlias>
</WaveRecipe>
```

Deleting a WaveRecipe Component

Use a simple destructiveChanges.xml file with only the WaveRecipe component declared. This deletes the WaveRecipe and any related WaveDataflow components. For more information, see Delete Components from an Organization. on page 66

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

Note: Use of the wildcard character doesn't return the recipe's associated dataflows.

WaveTemplateBundle

Represents an Analytics template bundle, which can be used to create Analytics apps. A bundle contains an Analytics template definition and all its related resources.This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

An Analytics template bundle is a folder that contains definition files for a template. Unlike other metadata components, a WaveTemplateBundle component isn't represented by a single component file, but instead by a collection of JSON and CSV definition files. Each definition file represents a resource in a template, such as lenses, dashboards, dataflows, and comma-separated values. For example, this directory structure shows the hierarchy of the folders and files for one Analytics Template definition, template1.

```
waveTemplates
    template1
        template-info.json
        variables.json
        ui.json
        extFiles
            PostalCodes.csv
```

Analytics template bundles must be under a top-level folder that's named `waveTemplates`. Each bundle must have its own subfolder under the `waveTemplates` folder and be named with the template's fully qualified API name. The bundle folder must contain a template-info.json file to specify the metadata about the template and the references to other definition files. An entire bundle doesn't have a suffix and definition files can have one of the following suffixes.

Suffix	Component Type
.json	JavaScript Object Notation
.csv	Comma-Separated Values

Version

WaveTemplateBundle components are available in API version 35.0 and later.

Special Access Rules

Definitions can be created in both managed and unmanaged packages.

Fields

Field Name	Field Type	Description
`assetIcon`	string	The icon to use by default for new Analytics apps based on this template. Valid values are `1.png` through `20.png`.
`description`	string	The specification of the template.
`label`	string	Required. The label of the template.
`templateType`	string	Required. The type of the template. Valid values are: • `App` • `Dashboard` • `Lens`

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

WaveXmd

Represents the WaveXmd object in the Analytics application. This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

WaveXmd components have the suffix `.xmd` and are stored in the `wave` folder.

Version

WaveXmd components are available in API version 39.0 and later.

Fields

Field Name	Field Type	Description
`application`	string	The name of the Analytics application the XMD is associated with. Available in API version 43.0 and later.
`dataset`	string	Required. Specifies the dataset associated with this XMD.
`datasetConnector`	string	The name of the connector source for the dataset.
`datasetFullyQualifiedName`	string	Specifies the fully qualified name of the dataset version associated with this XMD.
`dates`	WaveXmdDate	List of dates, with formatting information.
`dimensions`	WaveXmdDimension	List of dimensions, with formatting information.
`measures`	WaveXmdMeasure	List of measures, with formatting information.
`organizations`	WaveXmdOrganization	List of organizations, for multi-organization support.
`origin`	string	The origin of the dataset version.
`type`	string	The XMD type. Valid values are: • `System` • `User` • `Main` • `Asset` Available in API version 43.0 and later.
`waveVisualization`	string	The visualization behavior for Analytics assets. Valid values are: • `dashboard` • `lens` Available in API version 43.0 and later.

WaveXmdDate

WaveXmdDate represents a date.

Field	Field Type	Description
`alias`	string	Required. Alias of the Date column.
`compact`	boolean	Indicates whether the date is displayed in compact form (`true`) or not (`false`).

Field	Field Type	Description
dateFieldDay	string	The day field.
dateFieldEpochDay	string	The epoch day field.
dateFieldEpochSecond	string	The epoch second field.
dateFieldFiscalMonth	string	The fiscal month field.
dateFieldFiscalQuarter	string	The fiscal quarter field.
dateFieldFiscalWeek	string	The fiscal week field.
dateFieldFiscalYear	string	The fiscal year field.
dateFieldFullYear	string	The full year field.
dateFieldHour	string	The hour field.
dateFieldMinute	string	The minute field.
dateFieldMonth	string	The month field.
dateFieldQuarter	string	The quarter field.
dateFieldSecond	string	The second field.
dateFieldWeek	string	The week field.
dateFieldYear	string	The year field.
description	string	The description of the date column.
firstDayOfWeek	int	Required. Represents the first day of the week.
fiscalMonthOffset	int	Required. Offset number of months for the fiscal year in relation to the calendar year.
isYearEndFiscalYear	boolean	Indicates whether the year end is the fiscal year (`true`) or not (`false`).
label	string	The label of the date column.
showInExplorer	boolean	Indicates whether the date is displayed in the explorer (`true`) or not (`false`).
sortIndex	int	Required. The index value the system assigns to indicate where the item appears in a list.
type	string	Required. The type of date. Valid values are: • `Date`—A legacy date type. Available when the time zone isn't enabled. • `DateOnly`—A date type without an associated time. Available when the time zone is enabled. • `DateTime`—A date type that contains both date and time parts. Available when the time zone is enabled.

WaveXmdDimension

WaveXmdDimension represents a dimension.

Field	Field Type	Description
`conditionalFormatting`	WaveXmdFormattingProperty	The conditional formatting property for the dimension. Available in API version 43.0 and later.
`customActions`	WaveXmdDimensionCustomAction	Custom actions linked to this dimension.
`customActionsEnabled`	boolean	Indicates whether the dimension has custom actions enabled (`true`) or not (`false`).
`dateFormat`	string	The format used for a date that is a dimension.
`defaultAction`	string	The default action assigned to a dimension. An action for a dimension can be `openSfdcRecord`, `openActionsMenu`, `none`, or a valid API name with dot notation like `Global.LogACall` or `FeedItem.Post`.
`description`	string	The description of the dimension.
`field`	string	Required. The field name of the dimension (used in queries).
`fullyQualifiedName`	string	The fully qualified name of the dimension.
`imageTemplate`	string	The image template.
`isDerived`	boolean	Required. Indicates whether the dimension is derived (`true`) or not (`false`).
`isMultiValue`	boolean	Indicates whether the dimension is multi-value (`true`) or not (`false`).
`label`	string	The label for the dimension.
`linkTemplate`	string	The template for formatting a link.
`linkTemplateEnabled`	boolean	Indicates whether the dimension has link templates enabled (`true`) or not (`false`).
`linkTooltip`	string	The tooltip to be displayed for links.
`members`	WaveXmdDimensionMember	The member overrides for a dimension.
`origin`	string	The origin of this dimension.
`recordDisplayFields`	WaveXmdRecordDisplayLookup	Ordered list of dimensions. The list defines the default order in which to display the dimensions in the user interface.
`recordIdField`	string	The record ID for this dimension.
`recordOrganizationIdField`	string	The record organization ID for this dimension.
`salesforceActions`	WaveXmdDimensionSalesforceAction	Salesforce actions linked to this dimension.

Field	Field Type	Description
salesforceActionsEnabled	boolean	Indicates whether the dimension has Salesforce actions enabled (true) or not (false).
showDetailsDefaultFieldIndex	int	Default order in which to show the dimensions in the user interface.
showInExplorer	boolean	Indicates whether the dimension is displayed in the explorer (true) or not (false).
sortIndex	int	Required. The index value the system assigns to indicate where the item appears in a list.

WaveXmdFormattingProperty

WaveXmdFormattingProperty represents an XMD formatting property for conditional formatting.

Field	Field Type	Description
formattingBins	WaveXmdFormattingBin	The formatting bins for this property.
formattingPredicates	WaveXmdFormattingPredicate	The formatting predicates for this property.
property	string	Required. The property name.
referenceField	string	Required. The reference field for this property.
sortIndex	int	Required. The index value the system assigns to indicate where the item appears in a list.
type	string	Required. The property type.

WaveXmdFormattingBin

WaveXmdFormatttingBin represents an XMD formatting bin for conditional formatting.

Field	Field Type	Description
bin	string	Required. The formatting bin.
formatValue	string	Required. The format value for the bin.
label	string	Required. The label for the bin.
sortIndex	int	Required. The index value the system assigns to indicate where the item appears in a list.

WaveXmdFormattingPredicate

WaveXmdFormattingPredicate represents an XMD formatting predicate for conditional formatting.

Field	Field Type	Description
formatValue	string	Required. The format value for the predicate.
operator	string	Required. The operator for the predicate.
sortIndex	int	Required. The index value the system assigns to indicate where the item appears in a list.
value	string	Required. The value for the predicate.

WaveXmdDimensionCustomAction

WaveXmdDimensionCustomAction represents a custom action in a dimension.

Field	Field Type	Description
customActionName	string	Required. The name of this custom action.
enabled	boolean	Required. Indicates whether the action is enabled for a specific dimension (`true`) or not (`false`).
icon	string	The icon for the action.
method	string	The method for the action.
sortIndex	int	Required. The index value the system assigns to indicate where the item appears in a list.
target	string	The target for the action.
tooltip	string	The tooltip for the action.
url	string	The URL for the action.

WaveXmdDimensionMember

WaveXmdDimensionMember represents a dimension.

Field	Field Type	Description
color	string	The color for the member.
label	string	The label for the member. Note: Multi-line text isn't supported
member	string	Required. The member value.
sortIndex	int	Required. The index value the system assigns to indicate where the item appears in a list.

WaveXmdRecordDisplayLookup

WaveXmdDimensionRecordDisplayLookup represents a record display field.

Field	Field Type	Description
`recordDisplayField`	string	Required. The field to display.
`sortIndex`	int	Required. The index value the system assigns to indicate where the item appears in a list.

WaveXmdDimensionSalesforceAction

WaveXmdDimensionSalesforceAction represents an action in a dimension.

Field	Field Type	Description
`enabled`	boolean	Required. Indicates whether the action is enabled for a specific dimension (`true`) or not (`false`).
`salesforceActionName`	string	Required. The name of the action.
`sortIndex`	int	Required. The index value the system assigns to indicate where the item appears in a list.

WaveXmdMeasure

WaveXmdMeasure represents a measure.

Field	Field Type	Description
`conditionalFormatting`	WaveXmdFormattingProperty	The conditional formatting for the measure. Available in API version 43.0 and later.
`currencies`	WaveXmdMeasure[]	The list of currency formats for multiple currencies. Use this field to set the format for each currency used in the dataset.
`currencyCode`	String	The default currency code for the dataset.
`dateFormat`	string	The format used for a date that is a measure.
`description`	string	The description of the measure.
`field`	string	Required. The field name of the measure (used in queries).
`formatCustomFormat`	string	The original (XMD 1.1) format array as a single string.
`formatDecimalDigits`	int	The number of digits displayed after the decimal place.
`formatDecimalSeparator`	string	The custom separator for the decimal place. Available in API version 48.0 and later.
`formatIsNegativeParens`	boolean	Indicates whether to display negative numbers with parentheses, rather than a minus sign (`true`) or not (`false`).

Field	Field Type	Description
formatPrefix	string	The prefix placed before the field value.
formatSuffix	string	The suffix placed after the field value.
formatThousandsSeparator	string	The custom separator for the thousands place. Available in API version 48.0 and later.
formatUnit	string	The unit string for the measure. For example, 'cm'.
formatUnitMultiplier	double	The multiplier for the unit.
fullyQualifiedName	string	The fully qualified name of the measure.
isDerived	boolean	Required. Indicates whether the measure is derived (true) or not (false).
isMultiCurrency	boolean	Indicates whether multiple currencies are available for this dataset (true) or not (false).
label	string	The label for the measure.
origin	string	The origin of the measure.
showDetailsDefaultFieldIndex	int	Default order in which to show the measures in the user interface.
showInExplorer	boolean	Indicates whether the measure is displayed in the explorer (true) or not (false).
sortIndex	int	Required. The index value the system assigns to indicate where the item appears in a list.

WaveXmdOrganization

WaveXmdOrganization represents a Salesforce organization.

Field	Field Type	Description
instanceUrl	string	Required. The instance URL for the organization.
label	string	Required. The label for the organization.
organizationIdentifier	string	Required. The ID of the organization.
sortIndex	int	Required. The index value the system assigns to indicate where the item appears in a list.

Declarative Metadata Sample Definition

The following is an example of a WaveXmd component for an Asset XMD belonging to a dashboard.

```
<WaveXmd>
    <dataset xsi:nil="true"/>
```

```
    <dimesions>
        <conditionalFormatting>
            <formattingBins>
                <bin>*</bin>
                <formatValue>#FFFFFF</formatValue>
                <label xsi:nil="true"/>
                <sortIndex>0</sortIndex>
            </formattingBins>
            <formattingBins>
                <bin>0</bin>
                <formatValue>#000000</formatValue>
                <label xsi:nil="true"/>
                <sortIndex>1</sortIndex>
            </formattingBins>
            <property>chartColor</property>
            <referenceField>count</referenceField>
            <sortIndex xsi:nil="true"/>
            <type>multiple</type>
        </conditionalFormatting>
        <field>all_1.ALL</field>
        <isDerived>false</isDerived>
        <sortIndex>0</sortIndex>
    </dimensions>
    <measures>
        <conditionalFormatting>
            <formattingBins>
                <bin>*</bin>
                <formatValue>#FFFFFF</formatValue>
                <label xsi:nil="true"/>
                <sortIndex>0</sortIndex>
            </formattingBins>
            <formattingBins>
                <bin>0</bin>
                <formatValue>#000000</formatValue>
                <label xsi:nil="true"/>
                <sortIndex>1</sortIndex>
            </formattingBins>
            <property>chartColor</property>
            <referenceField>count</referenceField>
            <sortIndex xsi:nil="true"/>
            <type>multiple</type>
        </conditionalFormatting>
        <field>all_1.count</field>
        <formatCustomFormat>[&quot;#,###.##%&quot;,1]</formatCustomFormat>
        <isDerived>false</isDerived>
        <sortIndex>0</sortIndex>
    </measures>
    <type>Asset</type>
    <waveVisualization>dashboard</waveVisualization>
</WaveXmd>
```

WebStoreBundle

For internal use only.

WebStoreTemplate

Represents a configuration for creating commerce stores.

This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

WebStoreTemplate components have the suffix `.webstoretemplate` and are stored in the `webstoretemplate` folder.

Version

WebStoreTemplate components are available in API version 49.0 and later.

Special Access Rules

A B2B Commerce or D2C Commerce license and access to Commerce objects is required.

Fields

Field Name	Field Type	Description
`autoFacetingEnabled`	boolean	Indicates whether auto faceting is enabled (`true`) or not (`false`). If enabled (True), the most relevant search facets are automatically returned, in addition to the configured search facets, in the product search results. If disabled (False), only the configured search facets are returned. The default is `False`. See Add Product Search Filters (Facets) for more information. This field is available in API version 50.0 or later.
`cartAsyncProcessingEnabled`	boolean	Indicates whether add-to-cart requests are processed asynchronously (`True`) or not (`False`). The default value is `True`. This field is available in API version 59.0 or later.
`cartCalculateEnabled`	boolean	Indicates whether the cart calculate extension is enabled (`True`) or not (`False`). The default value is `False`. This field is available in API version 59.0 or later.
`cartToOrderAutoCustomFieldMapping`	boolean	Indicates whether custom field mapping for cart and order objects is enabled (`True`) or not (`False`). The default value is `True`. This field is available in API version 57.0 or later.

Field Name	Field Type	Description
checkoutTimeToLive	int	Amount of time in minutes that a checkout stays active and doesn't expire. This field is available in API version 52.0 and later.
checkoutValidAfterDate	dateTime	A timestamp in the default server timezone (GMT). All checkouts that start before this date are considered expired. This field is available in API version 52.0 and later.
commerceEinsteinActivitiesTracked	boolean	Indicates whether Commerce Einstein activities tracking is enabled (`true`) or not (`false`).
commerceEinsteinDeployed	boolean	Indicates whether Commerce Einstein is deployed (`true`) or not (`false`).
country	string	Two-digit ISO code of the store's country. Purchases can be shipped only to the country assigned to the store. Valid for only D2C stores. This field is available in API version 56.0 and later.
defaultCurrency	string	The template's default currency setting for new records.
defaultLanguage	string	Required. The template's default language setting for new records.
defaultTaxLocaleType	TaxLocaleType (enumeration of type string)	Required. The template's default tax type for your webstore. Possible values include: • `Automatic` • `Gross` • `Net`
description	string	The description of the template.
duplicateCartItemsEnabled	boolean	Indicates whether a cart can include multiple items with the same product ID (`True`) or not (`False`). The default value is `False`. This field is available in API version 59.0 or later.
guestBrowsingEnabled	boolean	Indicates whether guest browsing is enabled for this store. Set the option to `True` to allow guest buyers access to products in the store.
guestCartEnabled	boolean	Required. Indicates whether guest cart access is enabled for a store created with an LWR template. Set the option to `True` to allow guest buyers access to products in the store. This field is available in API version 58.0 and later.
guestCheckoutEnabled	boolean	Required. Indicates whether guest checkout access is enabled for a store created with an LWR template. Set the option to `True` to allow guest buyers access to products in the store. This field is available in API version 58.0 and later.
masterLabel	string	Required. The original (untranslated) name of a label. Each translated label is paired with its original untranslated version.
maxValuesPerFacet	int	Maximum number of values that can be added to a facet.

Field Name	Field Type	Description
orderActivationStatus	string	Status of the order. Possible values include: • Activated • Draft This field is available in API version 55.0 and later.
orderLifeCycleType	OrderLifeCycleType (enumeration of type string)	The order life cycle type. Possible values include: • MANAGED • UNMANAGED This field is available in API version 55.0 and later.
paginationSize	int	Number of results displayed per search results page.
preserveGuestCartEnabled	boolean	Required. Indicates whether cart contents are preserved when a guest logs in to the store. Set the option to True to preserve guest carts. This field is available in API version 60.0 and later.
pricingStrategy	PricingStrategy (enumeration of type string)	Required. The price selected to display to buyers. Possible values include: • LowestPrice • Priority The default value is LowestPrice.
productGrouping	ProductGrouping (enumeration of type string)	Determines whether product variations are listed individually in search results or are represented by the parent product, which links to its children. Possible values are: • NoGrouping—Variations are listed individually in search results. • VariationParent—The parent product is returned in search results with a link to its children. The default value is VariationParent. This field is available in API version 52.0 and later.
skipAdditionalEntitlementCheckForSearch	boolean	By default, user entitlement checks are run as part of a search index rebuild and again when product search results are returned. Skips the second check to promote faster search performance. Set the option to True to skip additional entitlement checks on a search. This field is available in API version 52.0 and later.
skuDetectionEnabled	boolean	Indicates whether SKU detection is enabled (true) or not (false).
splitShipmentEnabled	boolean	Required. Indicates whether split shipments are enabled (true) or not (false).
supportedCurrencies	string	Currencies supported for store template.
supportedLanguages	string	Required. Languages supported for store template.
supportedShipToCountries	string	Countries that a store created from the template can ship to.

Field Name	Field Type	Description
type	WebStoreType (enumeration of type string)	Required. The type of store configuration, B2C, B2B, or B2CE. Default is B2B.

Declarative Metadata Sample Definition

The following is an example of a web store template component.

```
<?xml version="1.0" encoding="UTF-8"?>
<WebStoreTemplate xmlns="http://soap.sforce.com/2006/04/metadata">
    <autoFacetingEnabled>true</autoFacetingEnabled>
    <cartAsyncProcessingEnabled>true</cartAsyncProcessingEnabled>
    <cartCalculateEnabled>false</cartCalculateEnabled>
    <cartToOrderAutoCustomFieldMapping>true</cartToOrderAutoCustomFieldMapping>
    <checkoutTimeToLive>10</checkoutTimeToLive>
    <checkoutValidAfterDate>2020-08-10T09:26:50</checkoutValidAfterDate>
    <commerceEinsteinActivitiesTracked>false</commerceEinsteinActivitiesTracked>
    <commerceEinsteinDeployed>false</commerceEinsteinDeployed>
    <country>US</country>
    <defaultCurrency>USD</defaultCurrency>
    <defaultLanguage>ENGLISH</defaultLanguage>
    <defaultTaxLocaleType>Net</defaultTaxLocaleType>
    <description>WebStore description</description>
    <duplicateCartItemsEnabled>false</duplicateCartItemsEnabled>
    <guestBrowsingEnabled>true</guestBrowsingEnabled>
    <guestCartEnabled>false</guestCartEnabled>
    <guestCartTimeToLive>10</guestCartTimeToLive>
    <guestCheckoutEnabled>false</guestCheckoutEnabled>
    <masterLabel>WebStore</masterLabel>
    <maxValuesPerFacet>99</maxValuesPerFacet>
    <orderActivationStatus>Activated</orderActivationStatus>
    <orderLifeCycleType>MANAGED</orderLifeCycleType>
    <paginationSize>9</paginationSize>
    <preserveGuestCartEnabled>false</preserveGuestCartEnabled>
    <pricingStrategy>Priority</pricingStrategy>
    <productGrouping>VariationParent</productGrouping>
   <skipAdditionalEntitlementCheckForSearch>true</skipAdditionalEntitlementCheckForSearch>

    <skuDetectionEnabled>false</skuDetectionEnabled>
    <supportedCurrencies>USD</supportedCurrencies>
    <supportedLanguages>en_us</supportedLanguages>
    <supportedShipToCountries>CA;US</supportedShipToCountries>
    <splitShipmentEnabled>false</splitShipmentEnabled>
    <type>B2B</type>
</WebStoreTemplate>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
```

```
        <members>*</members>
        <name>WebStoreTemplate</name>
    </types>
    <version>60.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

Workflow

Represents the metadata associated with a workflow rule. A workflow rule sets workflow actions into motion when its designated conditions are met. You can configure workflow actions to execute immediately when a record meets the conditions in your workflow rule, or set time triggers that execute the workflow actions on a specific day. Use this metadata type to create, update, or delete workflow rule definitions.

For more information, see Workflow in Salesforce Help. This type extends the Metadata metadata type and inherits its `fullName` field.

When using a manifest file, retrieve all workflow components using this code.

```
<types>
    <members>*</members>
    <name>Workflow</name>
</types>
```

Declarative Metadata File Suffix and Directory Location

Workflow files have the suffix `.workflow`. There's one file per standard or custom object that has workflow. These files are stored in the `workflows` directory of the corresponding package.

Version

Workflow rules are available in API version 13.0 and later.

Workflow

This metadata type represents the valid types of workflow rules and actions associated with a standard or custom object.

Field Name	Field Type	Description
`alerts`	WorkflowAlert[]	An array of all alerts for the object associated with the workflow.
`fieldUpdates`	WorkflowFieldUpdate[]	An array of all field updates for the object associated with the workflow.

Field Name	Field Type	Description
`flowActions`	WorkflowFlowAction[]	An array of flow triggers for the object associated with the workflow. Available in API version 30.0 and later. The pilot program for flow trigger workflow actions is closed. If you already enabled the pilot in your org, you can continue to create and edit flow trigger workflow actions. If you didn't enable the pilot, use Flow Builder to create a record-triggered flow, or use Process Builder to launch a flow from a process.
`fullName`	string	The developer name used as a unique identifier for API access. The `fullName` can contain only underscores and alphanumeric characters. It must be unique, begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. This field is inherited from the Metadata component.
`knowledgePublishes`	WorkflowKnowledgePublish[]	An array of Salesforce Knowledge Workflow Publishes associated with the workflow. Available in API version 27.0 and later.
`outboundMessages`	WorkflowOutboundMessage[]	An array of all the outbound messages for the object associated with the workflow.
`rules`	WorkflowRule[]	An array of all the objects associated with the workflow.
`tasks`	WorkflowTask[]	An array of all the tasks for the object associated with the workflow.

WorkflowActionReference

WorkflowActionReference represents one of the workflow actions.

Field Name	Field Type	Description
`name`	string	Required. The name of the workflow action.
`type`	WorkflowActionType (enumeration of type string)	Required. Available types of workflow actions: • `Alert` • `FieldUpdate` • `FlowAction`—Available in API version 30.0 and later • `OutboundMessage` • `Task` The pilot program for flow trigger workflow actions is closed. If you already enabled the pilot in your org, you can continue to create and edit flow trigger workflow actions. If you didn't enable the pilot, use Flow Builder to create a record-triggered flow, or use Process Builder to launch a flow from a process.

WorkflowAlert

WorkflowAlert represents an email alert associated with a workflow rule.

Field Name	Field Type	Description
`ccEmails`	string[]	Additional email addresses. This field is similar to the CC field in email clients. For the email to be sent successfully, set a value for `ccEmails` or `recipients`. You can set values for both fields. The value of `ccEmails` can include up to 5 different email addresses.
`description`	string	Required. A description of the email alert. Available in API version 16.0 and later.
`fullName`	string	Required. The developer name used as a unique identifier for API access. The `fullName` can contain only underscores and alphanumeric characters. It must be unique, begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. This field is inherited from the Metadata component.
`protected`	boolean	Required. Indicates whether this component is protected (`true`) or not (`false`). Protected components can't be linked to or referenced by components created in the installing organization.
`recipients`	WorkflowEmailRecipient[]	The recipients for the email. For the email to be sent successfully, set a value for `ccEmails` or `recipients`. You can set values for both fields.
`senderAddress`	string	The address in the From field for the email alert. With this address, you can use a standard global email address for your organization, such as `support@company.com`, instead of the default From field, which is the email address of the person who updates the record. You can only specify a value in this field if the `senderType` is set to `OrgWideEmailAddress`. See Organization-Wide Email Addresses in Salesforce Help.
`senderType`	ActionEmailSenderType (enumeration of type string)	The email used as the sender's From and Reply-To addresses. These values are valid. • `CurrentUser`—The email address of the person updating the record. This value is the default setting. • `DefaultWorkflowUser`—The email address of the default workflow user. If the email alert is installed from a package, this field value is changed to `CurrentUser`.

Field Name	Field Type	Description
		• `OrgWideEmailAddress`—A verified global email address for your organization, such as `support@company.com`.
`template`	string	Required. Named reference to an EmailTemplate. This email template isn't required to exist in the zip file, but it must exist in Metadata API. Lightning email templates aren't packageable. We recommend using a Classic email template.

WorkflowEmailRecipient

WorkflowEmailRecipient represents a recipient for an email alert associated with a workflow rule.

Field Name	Field Type	Description
`field`	string	Name of the field referenced in `type`. The field named is of the type specified in `type`.
`recipient`	string	The recipients for the email. Depending on the type selected, this field is required.
`type`	ActionEmailRecipientTypes (enumeration of type string)	Named reference to an EmailTemplate component. Valid values are: • `accountOwner`—The email is sent to the record's account owner. For example, the Account owner for an Opportunity. • `accountTeam`—Only applicable on the Account object. The email is sent to everyone on that Account's account team. • `campaignMemberDerivedOwner`—Emails are sent to lead and contact owners when contacts are added to a campaign or in response to a campaign. • `contactLookup`—The email is sent to a contact whose value is looked up from a field on the record. For this value, the `field` field must reference a Contact. • `creator`—The email is sent to the record's creator. • `customerPortalOwner`—The email is sent to a specific self-service portal user. For this value, the recipient field must reference a self-service portal user by their username. • `email`—The email is sent to an email address whose value is looked up from a field on the record. For this value, the `field` field must reference an email field.

Field Name	Field Type	Description
		• `group`—The email is sent to all users in a group. For this value, the recipient field must reference a group by group name. • `opportunityTeam`—Only applicable on the Opportunity object. The email is sent to everyone on that Opportunity's opportunity team. • `owner`—The email is sent to the record's owner. • `partnerUser`—The email is sent to a specific partner user. For this value, the recipient field must reference a partner user by username. • `portalRole` - Like `role`, but for portal roles only. • `portalRoleSubordinates` - Like `roleSubordinates`, but for portal roles only. • `role`—The email is sent to all users in a role. For this value, the recipient field must reference a role name in the role hierarchy. • `roleSubordinates`—The email is sent to all users in a role subordinate. For this value, the recipient field must reference a role. • `roleSubordinatesInternal`—Like `roleSubordinates`, but for internal portal roles only. • `user`—The email is sent to a specific user. For this value, the recipient field must reference a user by username. • `userLookup`—The email is sent to a user whose value is looked up from a field on the record. For this value, the `field` field must reference a user foreign key field.

WorkflowFieldUpdate

WorkflowFieldUpdate represents a workflow field update. With field updates, you can automatically update a field value to one that you specify when a workflow rule is triggered.

Field Name	Field Type	Description
`description`	string	The description of the field update. This information is useful to track the reasoning for initially configuring the field update.
`field`	string	Required. The field on the object for the workflow to be updated.
`formula`	string	If the `operation` field value is `Formula`, the formula used to compute the new field value.
`fullName`	string	Required. The developer name used as a unique identifier for API access. The `fullName` can contain only underscores and alphanumeric characters. It must be unique, begin with a letter,

Field Name	Field Type	Description
		not include spaces, not end with an underscore, and not contain two consecutive underscores. This field is inherited from the Metadata component.
`literalValue`	string	If the `operation` field value is `Literal`, the literal value for the field.
`lookupValue`	string	If the `operation` field value is `lookupValue`, the lookup value that is referenced.
`lookupValueType`	LookupValueType (enumeration of type string)	The type of object that the `lookupValue` field value is referencing. The valid values are: • `Queue` • `RecordType` • `User`
`name`	string	Required. A name for the component. Available in version API 16.0 and later.
`notifyAssignee`	boolean	Required. Notify the assignee when the field is updated.
`operation`	FieldUpdateOperation (enumeration of type string)	Required. The operation that computes the value with which to update the field. Valid values are: • `Formula`—Indicates the field is set to a formula. If set, the formula must be a valid formula. • `Literal`—Indicates the field is set to a literal value. If set, the literalValue must be a valid literal value for this field. • `LookupValue`—Similar to Literal, but for an object reference, such as a contact, user, or account. If set, the `lookupValue` element must be set. Only User is supported in the current API. • `NextValue`—Indicates that the field will be set to its next value. Only allowed when the field update references a picklist. • `Null`—Indicates that the field is set to null. • `PreviousValue`—Indicates that the field is set to its previous value. Only allowed when the field update references a picklist.
`protected`	boolean	Required. Indicates whether this component is protected (`true`) or not (`false`). Protected components can't be linked to or referenced by components created in the installing organization.
`reevaluateOnChange`	boolean	When set to `true`, if the field update changes the field's value, all workflow rules on the associated object are reevaluated. Any workflow rules whose criteria are met as a result of the field value change are triggered.

Field Name	Field Type	Description
		If any of the triggered workflow rules result in another field update that's also enabled for workflow rule reevaluation, a domino effect occurs, and more workflow rules can be reevaluated as a result of the newly triggered field update. This cascade of workflow rule reevaluation and triggering can happen up to 5 times after the initial field update that started it.
`targetObject`	string	Object set if the change is detected on a child record. If set, the object points to the foreign key reference on the child object that points to the parent. For example, if `EmailMessage` child record is changed, `EmailMessage.ParentId` points to the `Case` parent. This field is named `sourceField` before version 14.0. The field name change is automatically handled between versions and doesn't require any manual editing of existing XML component files.

WorkflowFlowAction

Represents a flow trigger, which is a workflow action that launches a flow. Available in API version 30.0 and later. For more information, see these topics in Salesforce Help.

- Define a Flow Trigger for Workflow (Pilot)
- Flow Trigger Considerations (Pilot)

Note:

- The pilot program for flow trigger workflow actions is closed. If you already enabled the pilot in your org, you can continue to create and edit flow trigger workflow actions. If you didn't enable the pilot, use Flow Builder to create a record-triggered flow, or use Process Builder to launch a flow from a process.
- Test mode for flow triggers isn't supported in the Metadata API. If you want a flow trigger to run the latest flow version when an administrator causes the workflow rule to fire, enable test mode via the user interface after deployment.

Field Name	Field Type	Description
`description`	string	Describes the flow trigger.
`flow`	string	Required. API name of the flow that this workflow action launches.
`flowInputs`	WorkflowFlowActionParameter[]	An array of values to pass into flow variables when launching the flow.
`label`	string	Required. Name of the flow trigger.
`language`	string	Reserved for future use.
`protected`	boolean	Reserved for future use.

WorkflowFlowActionParameter

Represents a value specified in the flow trigger that is passed into a variable when launching the flow.

Note: The pilot program for flow trigger workflow actions is closed. If you already enabled the pilot in your org, you can continue to create and edit flow trigger workflow actions. If you didn't enable the pilot, use Flow Builder to create a record-triggered flow, or use Process Builder to launch a flow from a process.

Field Name	Field Type	Description
`name`	string	Required. API name of the flow variable. The flow variable must have `isInput` set to `True`.
`value`	string	Required. Value to assign to the flow variable when launching the flow. If the variable's data type is sObject, `value` must be a merge field that identifies a record—or a lookup relationship field that references a record—of the same object type as the variable. For example: • *`{!this}`*—Identifies the record that fired the workflow rule. • *`{!Contact}`*—Identifies the contact associated with the record that fired the workflow rule. • *`{!Asset.Account}`*—Identifies the account associated with the asset that is associated with the record that fired the workflow rule. • *`{!SomeObject__r}`*—Uses a lookup relationship field to identify a custom object record associated with the record that fired the workflow rule. For variables of other data types, you can enter a merge field or a literal value. Manually enter a literal value when the variable requires the same value every time the flow is launched, regardless of which record fired the workflow rule. For example, you can enter *`true`* or *`false`* for a variable of type Boolean. Supported merge fields identify a global variable or a field of the same data type as the flow variable. For example: • *`{!Id}`*—ID of the record that fired the workflow rule. • *`{!Account.Owner.Email}`*—Email address of the account owner for the account associated with the record that fired the workflow rule. • *`{!$Organization.Country}`*—Country of the organization.

WorkflowKnowledgePublish

WorkflowKnowledgePublish represents Salesforce Knowledge article publishing actions and information. Available in API version 27.0 and later.

Field Name	Field Type	Description
`action`	KnowledgeWorkflowAction (enumeration of type string)	Required. The article publishing actions available when this rule fires. Valid values are: • `PublishAsNew`—Publishes the article as a new article.

Field Name	Field Type	Description
		• `Publish`—Publishes the article as a version of a previously published article.
`description`	string	A brief article description.
`label`	string	Required. Label that represents the article throughout the Salesforce user interface.
`language`	string	The language of the article.
`protected`	boolean	Required. Indicates whether this component is protected (`true`) or not (`false`). Protected components can't be linked to or referenced by components created in the installing organization.

WorkflowOutboundMessage

WorkflowOutboundMessage represents an outbound message associated with a workflow rule. Outbound messages are workflow and approval actions that send the information you specify to an endpoint you designate, such as an external service. An outbound message sends the data in the specified fields in the form of a SOAP message to the endpoint. For more information, see Outbound Message Actions in Salesforce Help.

Field Name	Field Type	Description
`apiVersion`	double	Required. The API version of the outbound message. Automatically set to the current API version when the outbound message is created. Valid API versions for outbound messages are 8.0 and 18.0 or later. This API version is used in API calls back to Salesforce using the enterprise or partner WSDLs. The `API Version` can only be modified by using Metadata API. It can't be modified using the Salesforce user interface. This field is available in API version 18.0 and later. If you change the `apiVersion` to a version that doesn't support one of the `fields` configured for the outbound message, the messages fail until you update your outbound message listener to consume the updated WSDL. To monitor the status of outbound messages, from Setup, in the Quick Find box, enter *`Outbound Messages`*, and then select **Outbound Messages** inSalesforce.
`description`	string	Describes the outbound message.
`endpointUrl`	string	Required. The endpoint URL to which the outbound message is sent.
`fields`	string[]	The named references to the fields to be sent.
`fullName`	string	Required. The developer name used as a unique identifier for API access. The `fullName` can contain only underscores and alphanumeric characters. It must be unique, begin with a letter, not include spaces, not

Field Name	Field Type	Description
		end with an underscore, and not contain two consecutive underscores. This field is inherited from the Metadata component.
`includeSessionId`	boolean	Required. Set if you want the Salesforce *session ID* included in the outbound message. Useful if you intend to make API calls and you don't want to include a username and password.
`integrationUser`	string	Required. The named reference to the user under which this message is sent.
`name`	string	Required. A name for the component. Available in version API 16.0 and later.
`protected`	boolean	Required. Indicates whether this component is protected (`true`) or not (`false`). Protected components can't be linked to or referenced by components created in the installing organization.
`useDeadLetterQueue`	boolean	This field is only available for organizations with dead letter queue permissions turned on. If set, this outbound message uses the dead letter queue if normal delivery fails.

WorkflowRule

This metadata type represents a workflow rule. This type extends the Metadata metadata type and inherits its `fullName` field.

Field Name	Field Type	Description
`actions`	WorkflowActionReference[]	An array of references for the actions that happen when this rule fires.
`active`	boolean	Required. Determines if this rule is active.
`booleanFilter`	string	For advanced criteria filter, the boolean formula. For example, `(1 AND 2) OR 3`.
`criteriaItems`	FilterItem[]	An array of the boolean criteria (conditions) under which this rule fires. Either `criteriaItems` or `formula` must be set.
`description`	string	The description of the workflow rule.
`failedMigrationToolVersion`	string	The API version in which a migration fails. Used as a reference to admins to retry the migration when the next version is released. Available in API version 54.0 and later.

Field Name	Field Type	Description
`formula`	string	The formula condition under which this rule first must be set, either `formula` or `criteriaItems`.
`fullName`	string	The developer name used as a unique identifier for API access. The `fullName` can contain only underscores and alphanumeric characters. It must be unique, begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. This field is inherited from the Metadata component.
`triggerType`	WorkflowTriggerTypes (enumeration of type string)	Under what conditions the trigger fires. Valid values are: • `onAllChanges`—The workflow rule is considered on all changes. • `onCreateOnly`—The workflow rule is considered only on create. • `onCreateOrTriggeringUpdate`—The workflow rule is considered on create and triggering updates.
`workflowTimeTriggers`	WorkflowTimeTrigger	Represents a set of Workflow actions, including Field Updates, Email Alerts, Outbound Messages, and Tasks, that executes before or after a specified interval of time.

WorkflowTask

This metadata type references an assigned workflow task.

Field Name	Field Type	Description
`assignedTo`	string	Specifies the user, role, or team to which the workflow rule or action is assigned. The field corresponding to the value specified here must be the same as the specified `assignedToType`.
`assignedToType`	ActionTaskAssignedToTypes (enumeration of type string)	Valid string values for this type are: • `accountCreator`—When set, the task is assigned to the record's account's creator. • `accountOwner`—When set, the task is assigned to the record's account owner (Opportunity).

Field Name	Field Type	Description
		• `accountTeam`—Same as WorkflowAlert type • `creator`—When set, the task is assigned to the record's creator. • `opportunityTeam`—Same as WorkflowAlert type • `owner`—When set, the task is assigned to the record's owner. • `partnerUser`—When set, the `assignedTo` field references a partner user by username. • `portalRole`—When set, the `assignedTo` field references a Role by role name, a portal role. • `role`—When set, the `assignedTo` field references a Role by role name. • `user`—When set, the `assignedTo` field references a User by username.
`description`	string	The description of this workflow task.
`dueDateOffset`	int	Required. The offset, in days, from either the trigger date, or the date specified in the (optional) `offsetFromField`. The offset can be a negative number.
`fullName`	string	Required. The developer name used as a unique identifier for API access. The `fullName` can contain only underscores and alphanumeric characters. It must be unique, begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. This field is inherited from the Metadata component.
`notifyAssignee`	boolean	Required. Set to send an email notification when the task is assigned.
`offsetFromField`	string	Optional field reference of the date field from which the `dueDate` is computed.
`priority`	string	Required. The priority to assign the created task.
`protected`	boolean	Required. Indicates whether this component is protected (`true`) or not (`false`). Protected components can't be linked to or referenced by components created in the installing organization.
`status`	string	Required. The status to assign the created task.
`subject`	string	Required. A subject for the workflow task that's used if an email notification is sent when the task is assigned. Available in API version 16.0 and later.

WorkflowTimeTrigger

Represents a set of Workflow actions, including Field Updates, Email Alerts, Outbound Messages, and Tasks, that execute before or after a specified interval of time.

Field Name	Field Type	Description
`actions`	WorkflowActionReference[]	An array of references for the actions that happen when this trigger fires.
`offsetFromField`	string	The date type field name that the time-based workflow triggers from, such as `Created Date`, `Last Modified Date`, `Rule Trigger Date`, or a custom date field on the object for which the workflow rule is defined.
`timeLength`	string	The numeric value of the time after or before the workflow triggers. A negative value represents the time length before the trigger fires. The `timeLength` is measured in either hours or days, as specified by `workflowTimeTriggerUnit`.
`workflowTimeTriggerUnit`	WorkflowTimeUnits (enumeration of type string)	The unit of time before or after which the time-based workflow triggers. Valid string values are: • `Hours` • `Days`

Declarative Metadata Sample Definition

Here's the definition of a workflow rule.

```
<?xml version="1.0" encoding="UTF-8"?>
<Workflow xmlns="http://soap.sforce.com/2006/04/metadata">
    <alerts>
        <fullName>Another_alert</fullName>
        <description>Another alert</description>
        <protected>false</protected>
        <recipients>
            <type>accountOwner</type>
        </recipients>
        <recipients>
            <field>Contact__c</field>
            <type>contactLookup</type>
        </recipients>
        <recipients>
            <field>Email__c</field>
            <type>email</type>
        </recipients>
        <template>TestEmail/Email Test</template>
    </alerts>
    <fieldUpdates>
        <fullName>Enum_Field_Update</fullName>
        <description>Blah</description>
        <field>EnumField__c</field>
```

```
        <name>Enum Field Update</name>
        <notifyAssignee>true</notifyAssignee>
        <operation>NextValue</operation>
        <protected>false</protected>
    </fieldUpdates>
    <fieldUpdates>
        <fullName>Enum_Field_Update2</fullName>
        <description>Blah</description>
        <field>EnumField__c</field>
        <literalValue>PLX2</literalValue>
        <name>Enum Field Update2</name>
        <notifyAssignee>true</notifyAssignee>
        <operation>Literal</operation>
        <protected>false</protected>
    </fieldUpdates>
    <fieldUpdates>
        <fullName>Field_Update</fullName>
        <description>TestField update desc</description>
        <field>Name</field>
        <formula>Name &amp;amp; &amp;quot;Updated&amp;quot;</formula>
        <name>Field Update</name>
        <notifyAssignee>false</notifyAssignee>
        <operation>Formula</operation>
        <protected>false</protected>
    </fieldUpdates>
    <fieldUpdates>
        <fullName>Lookup_On_Contact</fullName>
        <field>RealOwner__c</field>
        <lookupValue>admin@acme.com</lookupValue>
        <name>Lookup On Contact</name>
        <notifyAssignee>false</notifyAssignee>
        <operation>LookupValue</operation>
        <protected>false</protected>
    </fieldUpdates>
    <outboundMessages>
        <fullName>Another_Outbound_message</fullName>
        <description>Another Random outbound.</description>
        <endpointUrl>http://www.test.com</endpointUrl>
        <fields>Email__c</fields>
        <fields>Id</fields>
        <fields>Name</fields>
        <includeSessionId>true</includeSessionId>
        <integrationUser>admin@acme.com</integrationUser>
        <name>Another Outbound message</name>
        <protected>false</protected>
    </outboundMessages>
    <rules>
        <fullName>BooleanFilter</fullName>
        <active>false</active>
        <booleanFilter>1 AND 2 OR 3</booleanFilter>
        <criteriaItems>
            <field>CustomObjectForWorkflow__c.CreatedById</field>
            <operation>notEqual</operation>
        </criteriaItems>
```

```
        <criteriaItems>
            <field>CustomObjectForWorkflow__c.CreatedById</field>
            <operation>notEqual</operation>
            <value>abc</value>
        </criteriaItems>
        <criteriaItems>
            <field>CustomObjectForWorkflow__c.CreatedById</field>
            <operation>equals</operation>
            <value>xyz</value>
        </criteriaItems>
        <triggerType>onCreateOrTriggeringUpdate</triggerType>
    </rules>
    <rules>
        <fullName>Custom Rule1</fullName>
        <actions>
            <name>Another_alert</name>
            <type>Alert</type>
        </actions>
        <actions>
            <name>Enum_Field_Update2</name>
            <type>FieldUpdate</type>
        </actions>
        <actions>
            <fullName>Field_Update</name>
                <type>FieldUpdate</type>
        </actions>
        <actions>
            <name>Another_Outbound_message</name>
            <type>OutboundMessage</type>
        </actions>
        <actions>
            <name>Role_task_was_completed</name>
            <type>Task</type>
        </actions>
        <active>true</active>
        <criteriaItems>
            <field>CustomObjectForWorkflow__c.Name</field>
            <operation>startsWith</operation>
            <value>ABC</value>
        </criteriaItems>
        <description>Custom Rule1 desc</description>
        <triggerType>onCreateOrTriggeringUpdate</triggerType>
    </rules>
    <rules>
        <fullName>IsChangedFunctionRule</fullName>
        <active>true</active>
        <description>IsChangedDesc</description>
        <formula>ISCHANGED(Name)</formula>
        <triggerType>onAllChanges</triggerType>
    </rules>
    <tasks>
        <fullName>Another_task_was_completed</fullName>
        <assignedToType>owner</assignedToType>
        <description>Random Comment</description>
```

```
        <dueDateOffset>20</dueDateOffset>
        <notifyAssignee>true</notifyAssignee>
        <priority>High</priority>
        <protected>false</protected>
        <status>Completed</status>
        <subject>Another task was completed</subject>
    </tasks>
    <tasks>
        <fullName>Role_task_was_completed</fullName>
        <assignedTo>R11</assignedTo>
        <assignedToType>role</assignedToType>
        <dueDateOffset>-2</dueDateOffset>
        <notifyAssignee>true</notifyAssignee>
        <offsetFromField>CustomObjectForWorkflow__c.CreatedDate</offsetFromField>
        <priority>High</priority>
        <protected>false</protected>
        <status>Completed</status>
        <subject>Role task was completed</subject>
    </tasks>
    <tasks>
        <fullName>User_task_was_completed</fullName>
        <assignedTo>admin@acme.com</assignedTo>
        <assignedToType>user</assignedToType>
        <dueDateOffset>-2</dueDateOffset>
        <notifyAssignee>true</notifyAssignee>
        <offsetFromField>User.CreatedDate</offsetFromField>
        <priority>High</priority>
        <protected>false</protected>
        <status>Completed</status>
        <subject>User task was completed</subject>
    </tasks>
</Workflow>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

WorkSkillRouting

Represents a setup object that stores a set of WorkSkillRoutingAttribute objects. These objects are used to route a work item to an agent who has the skills necessary to take the work. This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

WorkSkillRouting components have the suffix `workSkillRouting` and are stored in the `workSkillRoutings` folder.

Version

WorkSkillRouting components are available in API version 46.0 and later.

Fields

Field Name	Field Type	Description
`isActive`	boolean	Required. Indicates whether assignment rules are active and can be evaluated.
`masterLabel`	string	Required. The label for this object. This display value is the internal label that is not translated.
`relatedEntity`	string	Required. Type of Salesforce object that the attributes are associated with.
`workSkillRoutingAttributes`	WorkSkillRoutingAttribute[]	A set of mappings between work-item field values and skills. Create one attribute mapping set for each object.

WorkSkillRoutingAttribute

Represents the routing assignments between object attributes and skills. Attributes are used to route a work item to an agent who has the skills necessary to take the work.

Fields

Field Name	Field Type	Description
`field`	string	Required. Field that this attribute applies to.
`isAdditionalSkill`	boolean	After a designated timeout period, additional skills are dropped from Omni-Channel routing. The case is then routed to the best-matched agent even if they don't have all the skills.
`skill`	string	Required. Skill used to route the work item when the attribute maps to the value selected.
`skillLevel`	int	Level of the skill required. This value can range from 0 to 10.
`skillPriority`	int	For additional skills, specify the order in which a skill is dropped if after the Drop Additional Skills Timeout on the routing configuration, no agent with that skill is available. Skills with a lower priority rank (9 or 10) are dropped first. Skills with a higher priority rank (0 or 1) are dropped last. Skills with the same priority value are dropped as a group. You can set skill priority using attribute setup for skills-based routing or Apex code. Available in API version 49.0 and later.
`value`	string	Attribute value that is assigned to the selected skill.

Declarative Metadata Sample Definition

The following is an example of a WorkSkillRouting component.

```
<?xml version="1.0" encoding="UTF-8"?>
<WorkSkillRouting xmlns="http://soap.sforce.com/2006/04/metadata">
  <isActive>true</isActive>
  <masterLabel>Attribute setup for skills-based routing for Case object</masterLabel>
  <relatedEntity>Case</relatedEntity>
  <workSkillRoutingAttributes>
    <field>Case.Origin</field>
    <isAdditionalSkill>false</isAdditionalSkill>
    <skill>Technical_Skill</skill>
    <skillLevel>3</skillLevel>
    <skillPriority>2</skillPriority>
    <value>Web</value>
  </workSkillRoutingAttributes>
</WorkSkillRouting>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

CHAPTER 13 Headers

Use headers in Metadata API calls to set options for each call.

AllOrNoneHeader
Indicates whether to roll back all metadata changes when some of the records in a call result in failures.

CallOptions
Specifies the API client identifier.

DebuggingHeader
Specifies that the deployment result contains the debug log output, and specifies the level of detail included in the log. The debug log contains the output of Apex tests that are executed as part of a deployment.

SessionHeader
Specifies the session ID that the login call returns. This session ID is used to authenticate all subsequent Metadata API calls.

`AllOrNoneHeader`

Indicates whether to roll back all metadata changes when some of the records in a call result in failures.

Version

This header is available in API version 34.0 and later.

Supported Calls

createMetadata(), updateMetadata(), upsertMetadata(), deleteMetadata()

Usage

If this header isn't used in API version 34.0 and later, by default a call can save a partial set of records (equivalent to `AllOrNoneHeader=false`)—the records that are processed successfully are saved and records that have failures aren't saved.

Fields

Field Name	Type	Description
`allOrNone`	boolean	Set to `true` to cause all metadata changes to be rolled back if any records in the call cause failures. Set to `false` to enable

Field Name	Type	Description
		saving only the records that are processed successfully when other records in the call cause failures.

Sample Code—Java

Add the `AllOrNoneHeader` to the metadata connection before you perform a call as follows:

```
metadataConnection.setAllOrNoneHeader(true);
```

This next example shows how to use the `AllOrNoneHeader` when creating two custom objects. Because the second custom object doesn't have the required `Name` field, the `create()` call can't create this custom object and rolls back the first custom object. The output is shown after this code sample.

```
import com.sforce.soap.metadata.*;
import com.sforce.soap.metadata.Error;
import com.sforce.ws.ConnectionException;

public class CallWithHeader {

    MetadataConnection metadataConnection = null;

    public static void main(String[] args) throws ConnectionException {
     CallWithHeader samples = new CallWithHeader();
        samples.createWithHeader();
    }

    public CallWithHeader() throws ConnectionException {
        metadataConnection = MetadataLoginUtil.login();
    }

    public void createWithHeader() throws ConnectionException {
        // Define two custom objects to be inserted.
        CustomObject co1 = new CustomObject();
        String name1 = "MyCustomObject1";
        co1.setFullName(name1 + "__c");
        co1.setDeploymentStatus(DeploymentStatus.Deployed);
        co1.setDescription("Created by the Metadata API");
        co1.setEnableActivities(true);
        co1.setLabel(name1 + " Object");
        co1.setPluralLabel(co1.getLabel() + "s");
        co1.setSharingModel(SharingModel.ReadWrite);

        CustomField nf = new CustomField();
        nf.setType(FieldType.Text);
        nf.setLabel(co1.getFullName() + " Name");
        co1.setNameField(nf);

        // The second custom object doesn't have a Name field
        CustomObject co2 = new CustomObject();
        String name2 = "MyCustomObject2";
        co2.setFullName(name2 + "__c");
```

```
        co2.setDeploymentStatus(DeploymentStatus.Deployed);
        co2.setDescription("Created by the Metadata API");
        co2.setEnableActivities(true);
        co2.setLabel(name2 + " Object");
        co2.setPluralLabel(co2.getLabel() + "s");
        co2.setSharingModel(SharingModel.ReadWrite);


        // Setting the allOrNone header to true to cause
        //  the call to not commit any record if one or more
        //  records in this call have failures.
        metadataConnection.setAllOrNoneHeader(true);

        // Now that the header has been set, make the create call.
        SaveResult[] results = metadataConnection
                .createMetadata(new Metadata[] { co1, co2 });

        // Iterate through the call results
        for (SaveResult r : results) {
            if (r.isSuccess()) {
                System.out.println("Created component: " + r.getFullName());
            } else {
                System.out
                        .println("Errors were encountered while creating "
                                + r.getFullName());
                for (Error e : r.getErrors()) {
                    System.out.println("Error message: " + e.getMessage());
                    System.out.println("Status code: " + e.getStatusCode());
                }
            }
        }
    }

}
```

This is the output that the sample returns. The first record is rolled back and the second has a failure.

```
Errors were encountered while creating MyCustomObject1__c
Error message: Record rolled back because not all records were valid and the request was
using AllOrNone header
Status code: ALL_OR_NONE_OPERATION_ROLLED_BACK
Errors were encountered while creating MyCustomObject2__c
Error message: Must specify a nameField of type Text or AutoNumber
Status code: FIELD_INTEGRITY_EXCEPTION
```

CallOptions

Specifies the API client identifier.

Version

This call is available in all API versions.

Supported Calls

All Metadata API calls.

Fields

Field Name	Type	Description
`client`	string	A value that identifies an API client.

Sample Code—Java

To change the API client ID, add the `CallOptions` header to the metadata connection before you perform a call as follows:

```
metadataConnection.setCallOptions("client ID");
```

DebuggingHeader

Specifies that the deployment result contains the debug log output, and specifies the level of detail included in the log. The debug log contains the output of Apex tests that are executed as part of a deployment.

Version

This header is available in all API versions.

Supported Calls

`deploy()`

Fields

Field Name	Type	Description
`categories`	LogInfo[]	A list of log categories with their associated log levels.
`debugLevel`	LogType (enumeration of type string)	Deprecated. This field is provided only for backward compatibility. If you provide values for both `debugLevel` and `categories`, the `categories` value is used. The `debugLevel` field specifies the type of information returned in the debug log. The values are listed from the least amount of information returned to the most information returned. Valid values include: • `None` • `Debugonly`

Field Name	Type	Description
		• Db • Profiling • Callout • Detail

LogInfo

Specifies the type and amount of information to be returned in the debug log. The `categories` field takes a list of these objects. LogInfo is a mapping of `category` to `level`.

Element Name	Type	Description
`category`	LogCategory	Specify the type of information returned in the debug log. Valid values are: • Db • Workflow • Validation • Callout • Apex_code • Apex_profiling • Visualforce • System • All
`level`	LogCategoryLevel	Specifies the level of detail returned in the debug log. Valid log levels are (listed from lowest to highest): • NONE • ERROR • WARN • INFO • DEBUG • FINE • FINER • FINEST

Sample Code—Java

Add the `DebuggingHeader` to the metadata connection before you perform the `deploy()` call as follows.

```
LogInfo[] logs = new LogInfo[1];
logs[0] = new LogInfo();
```

```
logs[0].setCategory(LogCategory.Apex_code);
logs[0].setLevel(LogCategoryLevel.Fine);
metadataConnection.setDebuggingHeader(logs);
```

The result of the `deploy()` call is obtained by calling `checkDeployStatus()`. After the deployment finishes, and if tests were run, the response of `checkDeployStatus()` contains the debug log output in the `debugLog` field of a `DebuggingInfo` output header.

SessionHeader

Specifies the session ID that the login call returns. This session ID is used to authenticate all subsequent Metadata API calls.

Version

This header is available in all API versions.

Supported Calls

All Metadata API calls.

Fields

Field Name	Type	Description
`sessionId`	string	The session ID that the login call returns.

Sample Code—Java

Add the `SessionHeader` to the metadata connection before you perform a call as follows:

```
metadataConnection.setSessionHeader("<session_ID>");
```

APPENDICES

APPENDIX A CustomObjectTranslation Language Support: Fully Supported Languages

Not every language supports all the possible values for the fields in CustomObjectTranslation. Use this appendix to determine which field values a language supports.

Note: Salesforce offers three levels of language support: fully supported languages, end-user languages, and platform-only languages. This appendix provides information only for fully supported languages.

Chinese (Simplified)

plural
false

Chinese (Traditional)

plural
false

Danish

article
None
Definite
Indefinite

gender
Feminine
Neuter

plural
true
false

Dutch

gender
Feminine

Neuter

plural

true

false

Finnish

caseType

Ablative

Adessive

Allative

Elative

Essive

Genitive

Illative

Inessive

Nominative

Partitive

Translative

plural

true

false

possessive

None

First

Second

French

gender

Masculine

Feminine

startsWith

Consonant

Vowel

plural

true

false

German

caseType
- Accusative
- Dative
- Genitive
- Nominative

gender
- Masculine
- Feminine
- Neuter

plural
- true
- false

Italian

gender
- Masculine
- Feminine

startsWith
- Consonant
- Special
- Vowel

plural
- true
- false

Japanese

plural
- false

Korean

plural
- false

Norwegian

article
- Definite
- Indefinite
- None

gender
- Masculine
- Feminine
- Neuter

plural
- true
- false

Portuguese (Brazil)

gender
- Masculine
- Feminine

plural
- true
- false

Russian

caseType
- Accusative
- Dative
- Genitive
- Instrumental
- Nominative
- Prepositional

gender
- Masculine
- Feminine
- Neuter
- AnimateMasculine

plural
- true

false

Spanish

gender

Masculine

Feminine

plural

true

false

Spanish (Mexico)

gender

Masculine

Feminine

plural

true

false

Swedish

article

None

Definite

Indefinite

gender

Feminine

Neuter

plural

true

false

Thai

plural

false

APPENDIX B CustomObjectTranslation Language Support: End-User Languages

Not every language supports all the possible values for the fields in CustomObjectTranslation. Use this appendix to determine which field values a language supports.

Note: Salesforce offers three levels of language support: fully supported languages, end-user languages, and platform-only languages. This appendix provides information only for end-user languages.

Arabic

article

Definite

None

gender

Masculine

Feminine

plural

true

false

possessive

None

First

Second

Bulgarian

gender

Masculine

Feminine

Neuter

plural

true

false

Croatian

caseType

Accusative

Dative

Genitive

Instrumental

Locative

Nominative

gender

Feminine

Masculine

Neuter

plural

true

false

Czech

caseType

Accusative

Dative

Genitive

Instrumental

Locative

Nominative

gender

Masculine

Feminine

Neuter

AnimateMasculine

plural

true

false

English (UK)

plural

false

true

startsWith

Consonant

Vowel

Greek

caseType

Accusative

Genitive

Nominative

gender

Masculine

Feminine

Neuter

plural

true

false

Hebrew

article

Definite

None

gender

Masculine

Feminine

plural

true

false

Hungarian

caseType

Ablative

Accusative

Allative

Causalfinal

Dative

Delative

Distributive

Elative

Essiveformal

Illative

Inessive

Instrumental

Nominative

Sublative

Termanative

Translative

Superessive

plural

true

false

possessive

None

First

Second

startsWith

Consonant

Vowel

Indonesian

plural

false

true

Polish

caseType

Nominative

Accusative

Dative

Genitive

Instrumental

Locative

gender

Masculine

Feminine

Neuter

AnimateMasculine

plural

true

false

Portuguese (Portugal)

gender

Feminine

Masculine

plural

true

false

Romanian

article

Definite

None

gender

Masculine

Feminine

Neuter

plural

true

false

Slovak

caseType

Accusative

Dative

Genitive

Instrumental

Nominative

Locative

gender

Feminine

Masculine

Neuter

AnimateMasculine

plural

true

false

Slovenian

caseType

Accusative

Dative

Genitive

Instrumental

Nominative

Locative

gender

Feminine

Masculine

Neuter

AnimateMasculine

plural

true

false

Turkish

caseType

Ablative

Accusative

Dative

Genitive

Nominative

Locative

possessive

None

First

Second

plural
- true
- false

Ukrainian

caseType
- Accusative
- Dative
- Genitive
- Instrumental
- Nominative
- Locative

gender
- Masculine
- Feminine
- Neuter
- AnimateMasculine

plural
- true
- false

Vietnamese

plural
- true
- false

APPENDIX C StandardValueSet Names and Standard Picklist Fields

In API version 38.0 and later, standard picklists are represented by the StandardValueSet type. In previous versions, standard picklists are represented by the CustomField type. This table lists the names of standard picklists as standard value sets and their corresponding field names.

Note: The names of standard value sets and picklist fields are case-sensitive.

Standard Value Set Name (API version 38.0 and later)	Field Name (API version 37.0 and earlier)
AccPlanObjectivePriority	AccountPlanObjective.Priority
AccPlanObjectiveStatus	AccountPlanObjective.Status
AAccreditationRating[1]	Accreditation.AccreditationRating[1]
AccountContactMultiRoles	AccountContactRelation.Roles
AccountContactRole	AccountContactRole.Role
AccountOwnership	Account.Ownership AccountCleanInfo.Ownership
AccountPlanStatus	AccountPlan.Status
AccountRating	Account.Rating Lead.Rating
AccountType	Account.Type
AccreditationAccreditingBody[1]	Accreditation.AccreditingBody[1]
AccreditationStatus[1]	Accreditation.Status[1]
AccreditationSubType[1]	Accreditation.SubType[1]
AccreditationType[1]	Accreditation.Type[1]
ACInitSumEmployeeType[4]	AntiCorruptionInitSum.EmployeeType[4]
ACInitSumInitiativeType[4]	AntiCorruptionInitSum.InitiativeType[4]
ACISumRecipientCategory[4]	AntiCorruptionInitSum.RecipientCategory[4]
ACorruptionInitSumCountry[4]	AntiCorruptionInitSum.Country[4]

Standard Value Set Name (API version 38.0 and later)	Field Name (API version 37.0 and earlier)
ACorruptionInitSumRegion[4]	AntiCorruptionInitSum.Region[4]
ActivityTimeEnum[1]	ActivityTiming.ActivityTime[1]
AdmissionSource[1]	CareRequest.AdmissionSource[1]
AdmissionType[1]	CareRequest.AdmissionType[1]
AFSIType[6]	ApplicationFormSellerItem.Type[6]
AFSIBuiltUpAreaUnit[6]	ApplicationFormSellerItem.BuiltUpAreaUnit[6]
AFSIConstructionStage[6]	ApplicationFormSellerItem.ConstructionStage[6]
AFSIIntendedUse[6]	ApplicationFormSellerItem.IntendedUse[6]
AFSIItemCondition[6]	ApplicationFormSellerItem.ItemCondition[6]
AllergyIntoleranceCategory[1]	AllergyIntolerance.Category[1]
AllergyIntoleranceSeverity[1]	AllergyIntolerance.Severity[1]
AllergyIntoleranceStatus[1]	AllergyIntolerance.Status[1]
AllergyIntoleranceType[1]	AllergyIntolerance.Type[1]
AllergyVerificationStatus[1]	AllergyIntolerance.VerificationStatus[1]
AppealRequestReasonType[1]	CareRequest.AppealRequestReasonType[1]
AppFormProductDisbursement[6]	ApplicationFormProduct.DisbursementMethod[6]
ApplicationActionItemType[6]	ApplicationActionItem.Type[6]
ApplicationActionItemStatus[6]	ApplicationActionItem.Status[6]
ApplicantPhoneType[6]	Applicant.PhoneType[6]
ApplicantStage[6]	Applicant.Stage[6]
ApplicantRole[6]	Applicant.Role[6]
ApplicantSalutation[6]	Applicant.Salutation[6]
ApplicantBusinessEntityType[6]	Applicant.BusinessEntityType[6]
ApplicationFormStage[6]	ApplicationForm.Stage[6]
ApplicationFormProductApplicantVisibleStatus[6]	ApplicationFormProduct.ApplicantVisibleStatus[6]
ApplicationFormIntakeChannelType[6]	ApplicationForm.IntakeChannelType[6]
ApplicationFormProductFeeType[6]	ApplicationFormProductFee.FeeType[6]
ApplicationFormProductLoanPurpose[6]	ApplicationFormProduct.LoanPurpose[6]
ApplicationFormProductLoanType[6]	ApplicationFormProduct.LoanType[6]
ApplicationFormProductOwnership[6]	ApplicationFormProduct.OwnershipType[6]

Standard Value Set Name (API version 38.0 and later)	Field Name (API version 37.0 and earlier)
ApplicationFormProductPartnerVisibleStatus[6]	ApplicationFormProduct.PartnerVisibleStatus[6]
ApplicationFormProductProposalStage[6]	ApplicationFormProductProposal.Stage[6]
ApplicationFormProductProposalSelectedBy[6]	ApplicationFormProductProposal.SelectedBy[6]
ApplicationFormProductProposalInterestRateType[6]	ApplicationFormProductProposal.InterestRateType[6]
ApplicationFormProductStage[6]	ApplicationFormProduct.Stage[6]
ApplicationFormTierType[6]	ApplicationForm.TierType[6]
ApprovedLevelOfCare[1]	CareRequest.ApprovedLevelOfCare[1]
APTCategoryEnum	ActionPlanTemplate.Category
APTSubcategoryEnum	ActionPlanTemplate.Subcategory
APTSourceTypeEnum	ActionPlanTemplate.SourceType
AQuestionQuestionCategory	AssessmentQuestion.QuestionCategory
AReasonAppointmentReason[1]	AppointmentReason.AppointmentReason[1]
AssessmentRating	Assessment.AssessmentRating
AssessmentStatus	Assessment.AssessmentStatus
AssetActionCategory	AssetAction.Category
AssetRelationshipType	AssetRelationship.RelationshipType
AssetStatus	Asset.Status
AssociatedLocationType	AssociatedLocation.Type
AuthorNoteRecipientType[1]	AuthorNote.RecipientType[1]
BarrierCodeType[1]	CareBarrierType.CodeType[1]
BCCertificationType[1]	BoardCertification.CertificationType[1]
BenefitProcessType[5]	BenefitType.ProcessType[5]
BLicenseJurisdictionType[1]	BusinessLicense.JurisdictionType[1]
BLicenseVerificationStatus[1]	BusinessLicense.VerificationStatus[1]
BoardCertificationStatus[1]	BoardCertification.Status[1]
BusinessLicenseStatus[1]	BusinessLicense.Status[1]
CampaignMemberStatus	CampaignMember.Status
CampaignStatus	Campaign.Status
CampaignType	Campaign.Type
CardType	CardPaymentMethod.CardType

Standard Value Set Name (API version 38.0 and later)	Field Name (API version 37.0 and earlier)
CareAmbulanceTransReason[1]	CareRequestItem.AmbulanceTransportReason[1]
CareAmbulanceTransType[1]	CareRequestItem.AmbulanceTransportType[1]
CareBarrierPriority[1]	CareBarrier.Priority[1]
CareBarrierStatus[1]	CareBarrier.Status[1]
CareBenefitVerifyRequestStatus[1]	CareBenefitVerifyRequest.Status[1]
CareDeterminantPriority[1]	CareDeterminant.Priority[1]
CareDeterminantTypeDomain[1]	CareDeterminantType.Domain[1]
CareDeterminantTypeType[1]	CareDeterminantType.Type[1]
CareEpisodeStatus[1]	CareEpisode.Status[1]
CareEpisodeType[1]	CareEpisode.Type[1]
CareItemStatus[1]	CareRequestItem.Status[1]
CareItemStatusReason[1]	CareRequestItem.StatusReason[1]
CareMetricTargetType[1]	CareMetricTarget.Type[1]
CareObservationCategory[1]	CareObservation.Category[1]
CareObservationStatus[1]	CareObservation.ObservationStatus[1]
CarePlanActivityStatus[1]	CarePlanActivity.Status[1]
CarePlanAuthorizationType[1]	CarePlan.AuthorizationType[1]
CarePlanDetailDetailType[1]	CarePlanDetail.DetailType[1]
CarePreauthItemLaterality[1]	CarePreauthItem.Laterality[1]
CarePreauthStatus[1]	CarePreauth.Status[1]
CareProgramEnrolleeStatus[1]	CareProgramEnrollee.Status[1]
CareProgramGoalPriority[1]	CareProgramGoal.Priority[1]
CareProgramGoalStatus[1]	CareProgramGoal.Status[1]
CareProgramProductStatus[1]	CareProgramProduct.Status[1]
CareProgramProviderRole[1]	CareProgramProvider.Role[1]
CareProgramProviderStatus[1]	CareProgramProvider.Status[1]
CareProgramStatus[1]	CareProgram.Status[1]
CareProgramTeamMemberRole[1]	CareProgramTeamMember.Role[1]
CareQuantityType[1]	CareRequestItem.QuantityType[1]
CareRegisteredDeviceStatus[1]	CareRegisteredDevice.Status[1]

Standard Value Set Name (API version 38.0 and later)	Field Name (API version 37.0 and earlier)
CareRequestExtensionAmbulanceTransportReason[1]	CareRequestExtension.AmbulanceTransportReason[1]
CareRequestExtensionAmbulanceTransportType[1]	CareRequestExtension.AmbulanceTransportType[1]
CareRequestExtensionNursingHomeResidentialStatus[1]	CareRequestExtension.NursingHomeResidentialStatus[1]
CareRequestExtensionRequestType[1]	CareRequestExtension.RequestType[1]
CareRequestExtensionServiceLevel[1]	CareRequestExtension.ServiceLevel[1]
CareRequestMemberGender[1]	CareRequest.MemberGender[1]
CareRequestMemberPrognosis[1]	CareRequest.MemberPrognosis[1]
CareRequestQuantityType[1]	CareRequest.QuantityType[1]
CareRequestReviewerStatus[1]	CareRequestReviewer.Status[1]
CareSpecialtySpecialtyType[1]	CareSpecialty.SpecialtyType[1]
CareSpecialtySpecialtyUsage[1]	CareSpecialty.SpecialtyUsage[1]
CareTaxonomyTaxonomyType[1]	CareTaxonomy.TaxonomyType[1]
CareTeamStatus[1]	CareTeam.Status[1]
CaseContactRole	CaseContactRole.Role
CaseEpisodeSubType[5]	CaseEpisode.Subtype[5]
CaseEpisodeType[5]	CaseEpisode.Type[5]
CaseOrigin	Case.Origin
CasePriority	Case.Priority
CaseReason	Case.Reason
CaseServicePlanStatus[1]	CarePlan.Status[1]
CaseStatus	Case.Status
CaseType	Case.Type
CBCoverageType[1]	CoverageBenefit.CoverageType[1]
CBenefitItemLimitTermType[1]	CoverageBenefitItemLimit.TermType[1]
CBItemLimitCoverageLevel[1]	CoverageBenefitItemLimit.CoverageLevel[1]
CBItemLimitNetworkType[1]	CoverageBenefitItemLimit.NetworkType[1]
CCFALineOfBusiness[1]	CategorizedCareFeeAgreement.LineofBusiness[1]
CCPAdditionalBenefits[4]	CrbnCreditProject.AdditionalBenefits[4]
CCProjectMitigationType[4]	CrbnCreditProject.MitigationType[4]
CCPStandardsAgencyName[4]	CrbnCreditProject.StandardsAgencyName[4]

Standard Value Set Name (API version 38.0 and later)	Field Name (API version 37.0 and earlier)
CCreditProjectProjectType[4]	CrbnCreditProject.ProjectType[4]
CDPresentOnAdmission[1]	CareDiagnosis.PresentOnAdmission[1]
CEIdentifierIdUsageType[1]	ClinicalEncounterIdentifier.IdUsageType[1]
CEncounterAdmissionSource[1]	ClinicalEncounter.AdmissionSource[1]
CEncounterCategory[1]	ClinicalEncounter.Category[1]
CEncounterDietPreference[1]	ClinicalEncounter.DietPreference[1]
CEncounterFacilityStatus[1]	ClinicalEncounterFacility.Status[1]
CEncounterServiceType[1]	ClinicalEncounter.ServiceType[1]
CEncounterSpecialCourtesy[1]	ClinicalEncounter.SpecialCourtesy[1]
CEncounterStatus[1]	ClinicalEncounter.Status[1]
CEpisodeDetailDetailType[1]	CareEpisodeDetail.DetailType[1]
ChangeRequestRelatedItemImpactLevel	ChangeRequestRelatedItem.ImpactLevel
ChangeRequestBusinessReason	ChangeRequest.BusinessReason
ChangeRequestCategory	ChangeRequest.Category
ChangeRequestImpact	ChangeRequest.Impact
ChangeRequestPriority	ChangeRequest.Priority
ChangeRequestRiskLevel	ChangeRequest.RiskLevel
ChangeRequestStatus	ChangeRequest.Status
ClassRankReportingFormat[1]	PersonEducation.ClassRankReportingFormat[1]
ClassRankWeightingType[1]	PersonEducation.ClassRankWeightingType[1]
ClinicalAlertCategories[1]	ClinicalAlert.Categories[1]
ClinicalAlertStatus[1]	ClinicalAlert.Status[1]
ClinicalCaseType[1]	CareRequest.ClinicalCaseType[1]
ClinicalDetectedIssueSeverityLevel[1]	ClinicalDetectedIssue.SeverityLevel[1]
ClinicalDetectedIssueStatus[1]	ClinicalDetectedIssue.Status[1]
COComponentValueType[1]	CareObservationComponent.ValueType[1]
COCValueInterpretation[1]	CareObservationComponent.ValueInterpretation[1]
CodeSetCodeSetType[1]	CodeSet.CodeSetType[1]
CommunicationChannel[1]	EngagementInteraction.CommunicationChannel[1]
CompanyRelationshipType[4]	Supplier.CompanyRelationshipType[4]

Standard Value Set Name (API version 38.0 and later)	Field Name (API version 37.0 and earlier)
ConsequenceOfFailure	Asset.ConsequenceOfFailure
ContactPointAddressType	ContactPointAddress.AddressType
ContactPointUsageType	ContactPointAddress.UsageType
ContactRequestReason	ContactRequest.RequestReason
ContactRequestStatus	ContactRequest.Status
ContactRole	OpportunityContactRole.Role
ContractContactRole	ContractContactRole.Role
ContractLineItemStatus[1]	ContractLineItem.Status[1]
ContractStatus	Contract.Status
COProcessingResult[1]	CareObservation.ProcessingResult[1]
COValueInterpretation[1]	CareObservation.ValueInterpretation[1]
CPAActivityType[1]	CarePlanActivity.ActivityType[1]
CPADetailDetailType[1]	CarePlanActivityDetail.DetailType[1]
CPADetailDetailType[1]	CarePlanActivityDetail.DetailType[1]
CPAdverseActionActionType[1]	CareProviderAdverseAction.ActionType[1]
CPAdverseActionStatus[1]	CareProviderAdverseAction.Status[1]
CPAgreementAgreementType[1]	ContractPaymentAgreement.AgreementType[1]
CPAgreementLineofBusiness[1]	ContractPaymentAgreement.LineofBusiness[1]
CPAProhibitedActivity[1]	CarePlanActivity.ProhibitedActivity[1]
CPDProblemPriority[1]	CarePlanDetail.ProblemPriority[1]
CPEligibilityRuleStatus[1]	CareProgramEligibilityRule.Status[1]
CPEnrolleeProductStatus[1]	CareProgramEnrolleeProduct.Status[1]
CPEnrollmentCardStatus[1]	CareProgramEnrollmentCard.Status[1]
CPFSpecialtySpecialtyRole[1]	CareProviderFacilitySpecialty.SpecialtyRole[1]
CPPRole[5]	CaseProceedingParticipant.Role[5]
CProgramProductAvailability[1]	CareProgramProduct.Availability[1]
CPTemplateProblemPriority[1]	CarePlanTemplateProblem.Priority[1]
CRDDrugAdministrationSetting[1]	CareRequestDrug.DrugAdministrationSetting[1]
CRDNameType[1]	CareRegisteredDevice.NameType[1]
CCRDPriority[1]	CareRequestDrug.Priority[1]

Standard Value Set Name (API version 38.0 and later)	Field Name (API version 37.0 and earlier)
CCRDPriority[1]	CareRequestDrug.Priority[1]
CRDRequestType[1]	CareRequestDrug.RequestType[1]
CRDStatus[1]	CareRequestDrug.Status[1]
CRDStatusReason[1]	CareRequestDrug.StatusReason[1]
CRECaseSubStatus[1]	CareRequestExtension.CaseSubStatus[1]
CRECaseSubStatus[1]	CareRequestExtension.CaseSubStatus[1]
CREDocumentAttachmentStatus[1]	CareRequestExtension.DocumentAttachmentStatus[1]
CREIndependentReviewDetermination[1]	CareRequestExtension.IndependentReviewDetermination[1]
CREPriorAuthRequestIdentifier[1]	CareRequestExtension.AuthorizationRefIdentifier[1]
CREPriorDischargeStatus[1]	CareRequestExtension.PriorDischargeStatus[1]
CREReopenRequestOutcome[1]	CareRequestExtension.ReopenRequestOutcome[1]
CREReopenRequestType[1]	CareRequestExtension.ReopenRequestType[1]
CRERequestOutcome[1]	CareRequestExtension.RequestOutcome[1]
CRIApprovedLevelOfCare[1]	CareRequestItem.ApprovedLevelOfCare[1]
CRIClinicalDetermination[1]	CareRequestItem.ClinicalDetermination[1]
CRICurrentLevelOfCare[1]	CareRequestItem.CurrentLevelOfCare[1]
CRIDeniedLevelOfCare[1]	CareRequestItem.DeniedLevelOfCare[1]
CRIModifiedLevelOfCare[1]	CareRequestItem.ModifiedLevelOfCare[1]
CRIPriority[1]	CareRequestItem.Priority[1]
CRIRequestedLevelOfCare[1]	CareRequestItem.RequestedLevelOfCare[1]
CRIRequestType[1]	CareRequestItem.RequestType[1]
CRReviewerReviewerType[1]	CareRequestReviewer.ReviewerType[1]
CSBundleUsageType[1]	CodeSetBundle.Type[1]
CServiceRequestIntent[1]	ClinicalServiceRequest.Type[1]
CServiceRequestPriority[1]	ClinicalServiceRequest.Priority[1]
CServiceRequestStatus[1]	ClinicalServiceRequest.Status[1]
CSRequestDetailDetailType[1]	ClinicalServiceRequestDetail.DetailType[1]
CurrentLevelOfCare[1]	CareRequest.CurrentLevelOfCare[1]
DChecklistItemStatus[1]	DocumentChecklistItem.Status[1]
DecisionReason[1]	CareRequest.DecisionReason[1]

Standard Value Set Name (API version 38.0 and later)	Field Name (API version 37.0 and earlier)
DeniedLevelOfCare[1]	CareRequest.DeniedLevelOfCare[1]
DiagnosisCodeType[1]	CareDiagnosis.CodeType[1]
DiagnosticSummaryCategory[1]	DiagnosticSummary.Category[1]
DiagnosticSummaryStatus[1]	DiagnosticSummary.Status[1]
DigitalAssetStatus	Asset.DigitalAssetStatus
DEInclSumDiversityType[4]	DivrsEquityInclSum.DiversityType[4]
DEInclSumEmployeeType[4]	DivrsEquityInclSum.EmployeeType[4]
DEInclSumEmploymentType[4]	DivrsEquityInclSum.EmploymentType[4]
DEInclSumGender[4]	DivrsEquityInclSum.Gender[4]
DEISumDiversityCategory[4]	DivrsEquityInclSum.DiversityCategory[4]
DischargeDiagnosisCodeType[1]	CareDiagnosis.DischargeCodeType[1]
DIssueDetailType[1]	ClinicalDetectedIssueDetail.DetailType[1]
DivrsEquityInclSumLocation[4]	DivrsEquityInclSum.Location[4]
DivrsEquityInclSumRace[4]	DivrsEquityInclSum.Race[4]
DrugClinicalDetermination[1]	CareRequestDrug.ClinicalDetermination[1]
DSDDocumentRelationType[1]	DiagnosticSummaryDetail.DocumentRelationType[1]
DSDocumentStage[1]	DiagnosticSummary.DocumentStage[1]
DSummaryDetailDetailType[1]	DiagnosticSummaryDetail.DetailType[1]
DSummaryUsageType[1]	DiagnosticSummary.UsageType[1]
ECTypeContactPointType[1]	EngagementChannelType.ContactPointType[1]
EducationLevel[1]	PersonEducation.EducationLevel[1]
EEligibilityCriteriaStatus[1]	EnrollmentEligibilityCriteria.Status[1]
EmploymentOccupation[1]	PersonEmployment.Occupation[1]
EmploymentStatus[1]	PersonEmployment.EmploymentStatus[1]
EngagementAttendeeRole[1]	EngagementAttendee.Role[1]
EngagementSentimentEnum[1]	EngagementInteraction.Sentiment[1]
EngagementStatusEnum[1]	EngagementInteraction.Status[1]
EngagementTypeEnum[1]	EngagementInteraction.Type[1]
EnrolleeOptOutReasonType[1]	CareProgramEnrollee.OptOutReasonType[1]
EnrollmentStatus[5]	BenefitAssignment.Status[5]

Standard Value Set Name (API version 38.0 and later)	Field Name (API version 37.0 and earlier)
EntitlementType	Entitlement.Type
EBSEmployeeBenefitType[4]	EmpBenefitSummary.EmployeeBenefitType[4]
EBSPercentageCalcType[4]	EmpBenefitSummary.PercentageCalcType[4]
EBSummaryBenefitUsage[4]	EmpBenefitSummary.BenefitUsage[4]
EBSummaryEmploymentType[4]	EmpBenefitSummary.EmploymentType[4]
EBSEmployeeBenefitType[4]	EmpBenefitSummary.EmployeeBenefitType[4]
EBSPercentageCalcType[4]	EmpBenefitSummary.PercentageCalcType[4]
EDemographicSumAgeGroup[4]	EmployeeDemographicSum.AgeGroup[4]
EDemographicSumGender[4]	EmployeeDemographicSum.Gender[4]
EDemographicSumRegion[4]	EmployeeDemographicSum.Region[4]
EDemographicSumReportType[4]	EmployeeDemographicSum.ReportType[4]
EDemographicSumWorkType[4]	EmployeeDemographicSum.WorkType[4]
EDevelopmentSumGender[4]	EmployeeDevelopmentSum.Gender[4]
EDSumEmployeeType[4]	EmployeeDevelopmentSum.EmployeeType[4]
EDSumEmploymentType[4]	EmployeeDemographicSum.EmploymentType[4]
EDSumProgramCategory[4]	EmployeeDevelopmentSum.ProgramCategory[4]
EPSumMarket[4]	EconomicPerformanceSum.Market[4]
EPSumPerformanceCategory[4]	EconomicPerformanceSum.PerformanceCategory[4]
EPSumPerformanceType[4]	EconomicPerformanceSum.PerformanceType[4]
EPSumRegion[4]	EconomicPerformanceSum.Region[4]
ERCompanyBusinessRegion[4]	EmssnRdctnCommitment.CompanyBusinessRegion[4]
ERCompanySector[4]	EmssnRdctnCommitment.CompanySector[4]
EReductionTargetTargetType[4]	EmssnReductionTarget.TargetType[4]
ERTargetOtherTargetKpi[4]	EmssnReductionTarget.OtherTargetKpi[4]
ERTTargetSettingMethod[4]	EmssnReductionTarget.TargetSettingMethod[4]
EventSubject	Event.Subject
EventType	Event.Type
FacilityRoomBedType[1]	CareRequest.FacilityRoomBedType[1]
FinalLevelOfCare[1]	CareRequest.FinalLevelOfCare[1]
FinanceEventAction	FinanceTransaction.EventAction

Standard Value Set Name (API version 38.0 and later)	Field Name (API version 37.0 and earlier)
FinanceEventType	FinanceBalanceSnapshot.EventType FinanceTransaction.EventType
FiscalYearPeriodName	Period.PeriodLabel
FiscalYearPeriodPrefix	FiscalYearSettings.PeriodPrefix
FiscalYearQuarterName	Period.QuarterLabel
FiscalYearQuarterPrefix	FiscalYearSettings.QuarterPrefix
ForecastingItemCategory[2]	n/a
FreightHaulingMode[4]	FrgtHaulingEmssnFctr.FreightHaulingMode[4] FrgtHaulingEnrgyUse.FreightHaulingMode[4]
FtprntAuditApprovalStatus[4]	Scope3CrbnFtprnt.AuditApprovalStatus[4] StnryAssetCrbnFtprnt.AuditApprovalStatus[4] VehicleAssetCrbnFtprnt.AuditApprovalStatus[4] WasteFootprint.AuditApprovalStatus[4]
FulfillmentStatus	FulfillmentOrder.Status
FulfillmentType	FulfillmentOrder.Type
GADetailDetailType[1]	GoalAssignmentDetail.DetailType[1]
GoalAssignmentProgressionStatus[1]	GoalAssignment.ProgressionStatus[1]
GoalAssignmentStatus[1]	GoalAssignment.Status[1]
GoalDefinitionCategory[1]	GoalDefinition.Category[1]
GoalDefinitionUsageType[1]	GoalDefinition.UsageType[1]
GovtFinancialAsstSumType[4]	GovtFinancialAsstSum.Type[4]
GpaWeightingType[1]	PersonEducation.GpaWeightingType[1]
GrievanceType[1]	CareRequest.GrievanceType[1]
HCFacilityLocationType[1]	HealthcareFacility.LocationType[1]
HcpCategory[1]	HealthCareProcedure.Category[1]
HcpCodeType[1]	HealthCareProcedure.CodeType[1]
HealthCareDiagnosisCategory[1]	HealthCareDiagnosis.Category[1]
HealthCareDiagnosisCodeType[1]	HealthCareDiagnosis.CodeType[1]
HealthCareDiagnosisGender[1]	HealthCareDiagnosis.Gender[1]
HealthcareProviderStatus[1]	HealthcareProvider.Status[1]

Standard Value Set Name (API version 38.0 and later)	Field Name (API version 37.0 and earlier)
HealthConditionDetailType[1]	HealthConditionDetail.DetailType[1]
HealthConditionSeverity[1]	HealthCondition.Severity[1]
HealthConditionStatus[1]	HealthCondition.ConditionStatus[1]
HealthConditionType[1]	HealthCondition.Type[1]
HealthDiagnosticStatus[1]	HealthCondition.DiagnosticStatus[1]
HFNetworkGenderRestriction[1]	HealthcareFacilityNetwork.GenderRestriction[1]
HFNetworkPanelStatus[1]	HealthcareFacilityNetwork.PanelStatus[1]
HPayerNetworkNetworkType[1]	HealthcarePayerNetwork.NetworkType[1]
HPayerNwkLineOfBusiness[1]	HealthcarePayerNetwork.LineofBusiness[1]
HPFGenderRestriction[1]	HealthcarePractitionerFacility.GenderRestriction[1]
HPFTerminationReason[1]	HealthcarePractitionerFacility.TerminationReason[1]
HProviderNpiNpiType[1]	HealthcareProviderNpi.NpiType[1]
HProviderProviderClass[1]	HealthcareProvider.ProviderClass[1]
HProviderProviderType[1]	HealthcareProvider.ProviderType[1]
HPSpecialtySpecialtyRole[1]	HealthcareProviderSpecialty.SpecialtyRole[1]
HSActionLogActionStatus[1]	HealthScoreActionLog.ActionStatus[1]
IaAppInStatus[1]	IndividualApplication.Status[1]
IaAuthCategory[1]	IndividualApplication.Category[1]
IaInternalStatus[1]	IndividualApplication.InternalStatus[1]
IAItemStatus[1]	IndividualApplicationItem.Status[1]
IARejectionReason[1]	IndividualApplication.RejectionReason[1]
IAServiceType[1]	IndividualApplication.ServiceType[1]
IdeaCategory[3]	IdeaTheme.Categories[3]
IdeaMultiCategory	Idea.Categories
IdeaStatus	Idea.Status
IdeaThemeStatus	IdeaTheme.Status
IdentifierIdUsageType[1]	Identifier.IdUsageType[1]
IFnolChannel[1]	InsurancePolicy.FnolChannel[1]
IncidentCategory	Incident.Category
IncidentImpact	Incident.Impact

Standard Value Set Name (API version 38.0 and later)	Field Name (API version 37.0 and earlier)
IncidentPriority	Incident.Priority
IncidentRelatedItemImpactLevel	IncidentRelatedItem.ImpactLevel
IncidentRelatedItemImpactType	IncidentRelatedItem.ImpactType
IncidentReportedMethod	Incident.ReportedMethod
IncidentStatus	Incident.Status
IncidentSubCategory	Incident.SubCategory
IncidentType	Incident.Type
IncidentUrgency	Incident.Urgency
Industry	Account.Industry AccountCleanInfo.Industry Lead.Industry LeadCleanInfo.Industry
InterventionCodeType[1]	CareInterventionType.CodeType[1]
IPCancelationReasonType[1]	InsurancePolicy.CancellationReasonType[1]
IPCBenefitPaymentFrequency[1]	InsurancePolicyCoverage.BenefitPaymentFrequency[1]
IPCCategory[1]	InsurancePolicyCoverage.Category[1]
IPCCategoryGroup[1]	InsurancePolicyCoverage.CategoryGroup[1]
IPCDeathBenefitOptionType[1]	InsurancePolicyCoverage.DeathBenefitOptionType[1]
IPCIncomeOptionType[1]	InsurancePolicyCoverage.IncomeOptionType[1]
IPCLimitRange[1]	InsurancePolicyCoverage.LimitRange[1]
IPolicyAuditTerm[1]	InsurancePolicy.AuditTerm[1]
IPolicyChangeSubType[1]	InsurancePolicy.ChangeSubtype[1]
IPolicyChangeType[1]	InsurancePolicy.ChangeType[1]
IPolicyChannel[1]	InsurancePolicy.RenewalChannel[1]
IPolicyPlanTier[1]	InsurancePolicy.PlanTier[1]
IPolicyPlanType[1]	InsurancePolicy.PlanType[1]
IPolicyPolicyType[1]	InsurancePolicy.PolicyType[1]
IPolicyPremiumCalcMethod[1]	InsurancePolicy.PremiumCalculationMethod[1]
IPolicyPremiumFrequency[1]	InsurancePolicy.PremiumFrequency[1]
IPolicyPremiumPaymentType[1]	InsurancePolicy.PremiumPaymentType[1]

Standard Value Set Name (API version 38.0 and later)	Field Name (API version 37.0 and earlier)
IPolicyStatus[1]	InsurancePolicy.Status[1]
IPolicySubStatusCode[1]	InsurancePolicy.Substatus[1]
IPolicyTerm[1]	InsurancePolicy.PolicyTerm[1]
IPolicyTransactionStatus[1]	InsurancePolicyTransaction.Status[1]
IPolicyTransactionType[1]	InsurancePolicyTransaction.Type[1]
IPOwnerPOwnerType[1]	InsurancePolicyOwner.PolicyOwnerType[1]
IPParticipantRole[1]	InsurancePolicyParticipant.Role[1]
IPPRelationshipToInsured[1]	InsurancePolicyParticipant.RelationshipToInsured[1]
LeadSource	Account.AccountSource CampaignMember.LeadSource Contact.LeadSource Lead.LeadSource Opportunity.LeadSource
LeadStatus	Lead.Status
LicenseClassType[1]	BusinessLicense.LicenseClass[1]
LineOfAuthorityType[1]	BusinessLicense.LineOfAuthority[1]
LocationType	Location.LocationType
LPEVerificationStatus[6]	PartyExpense.VerificationStatus[6]
LPIAppInCategory[1]	BusinessLicenseApplication.ApplicationCategory[1]
LPIAppInStatus[1]	BusinessLicenseApplication.Status[1]
LPIIncomeType[6]	PartyIncome.IncomeType[6]
LPIIncomeStatus[6]	PartyIncome.IncomeStatus[6]
LPIIncomeFrequency[6]	PartyIncome.IncomeFrequency[6]
LPIVerificationStatus[6]	PartyIncome.VerificationStatus[6]
MedicationCategoryEnum[1]	Medication.MedicationCategory[1]
MedicationDispenseMedAdministrationSettingCategory[1]	MedicationDispense.MedAdministrationSettingCategory[1]
MedicationDispenseStatus[1]	MedicationDispense.Status[1]
MedicationDispenseSubstitutionReason[1]	MedicationDispense.SubstitutionReason[1]
MedicationDispenseSubstitutionType[1]	MedicationDispense.SubstitutionType[1]
MedicationStatementStatus[1]	MedicationStatement.Status[1]
MedicationStatus[1]	Medication.Status[1]

Standard Value Set Name (API version 38.0 and later)	Field Name (API version 37.0 and earlier)
MedReviewRepresentativeType[1]	MedicationTherapyReview.SurrogateType[1]
MedTherapyReviewSubtype[1]	MedicationTherapyReview.ReviewSubtype[1]
MemberPlanPrimarySecondaryTertiary[1]	MemberPlan.PrimarySecondaryTertiary[1]
MemberPlanRelToSub[1]	MemberPlan.RelationshipToSubscriber[1]
MemberPlanStatus[1]	MemberPlan.Status[1]
MemberPlanVerificStatus[1]	MemberPlan.VerificationStatus[1]
MilitaryService	Individual.MilitaryService
ModifiedCareCodeType[1]	CareRequestItem.ModifiedCodeType[1]
ModifiedDiagnosisCodeType[1]	CareDiagnosis.ModifiedCodeType[1]
ModifiedDrugCodeType[1]	CareRequestDrug.ModifiedCodeType[1]
ModifiedLevelOfCare[1]	CareRequest.ModifiedLevelOfCare[1]
MRequestPriority[1]	MedicationRequest.Priority[1]
MRequestStatus[1]	MedicationRequest.Status[1]
MRequestTherapyDuration[1]	MedicationRequest.TherapyDuration[1]
MRequestType[1]	MedicationRequest.Type[1]
MStatementDeliverySetting[1]	MedicationStatement.DeliverySetting[1]
MStatementDetailType[1]	MedicationStatementDetail.DetailType[1]
OcrService[1]	OcrDocumentScanResult.OcrService[1]
OcrStatus[1]	OcrDocumentScanResult.OcrStatus[1]
OIncidentSummaryHazardType[4]	OrgIncidentSummary.HazardType[4]
OISCorrectiveActionType[4]	OrgIncidentSummary.CorrectiveActionType[4]
OISummaryIncidentSubtype[4]	OrgIncidentSummary.IncidentSubtype[4]
OISummaryIncidentType[4]	OrgIncidentSummary.IncidentType[4]
OISummaryPenaltyType[4]	OrgIncidentSummary.PenaltyType[4]
OpportunityCompetitor	OpportunityCompetitor.CompetitorName
OpportunityStage	Opportunity.StageName
OpportunityType	Opportunity.Type
OrderItemSummaryChgRsn	OrderItemSummaryChange.Reason
OrderStatus	Not available in 37.0 and earlier
OrderSummaryRoutingSchdRsn	OrderSummaryRoutingSchedule.Reason

Standard Value Set Name (API version 38.0 and later)	Field Name (API version 37.0 and earlier)
OrderSummaryStatus	OrderSummary.Status
OrderType	Order.Type
ParProvider[1]	CareRequest.ParProvider[1]
PERecurrenceInterval[6]	PartyExpense.RecurrenceInterval[6]
PartyExpenseStatus[6]	PartyExpense.Status[6]
PartyExpenseType[6]	PartyExpense.Type[6]
PartnerRole	PartnerRole.ReverseRole
PartyProfileCountryofBirth	PartyProfile.CountryofBirth
PartyProfileEmploymentType	PartyProfile.EmploymentType
PartyProfileFundSource	PartyProfile.FundSource
PartyProfileGender	PartyProfile.Gender
PartyProfileResidentType	PartyProfile.ResidentType
PartyProfileReviewDecision	PartyProfile.ReviewDecision
PartyProfileRiskType	PartyProfile.RiskType
PartyProfileStage	PartyProfile.Stage
PartyScreeningStepType	PartyScreeningStep.Type
PartyScreeningSummaryStatus	PartyScreeningSummary.Status
PatientImmunizationStatus[1]	PatientImmunization.Status[1]
PaymentMandateAccountType[6]	PaymentMandate.BankAccountType[6]
PaymentMandateMandateFrequency[6]	PaymentMandate.MandateFrequency[6]
PaymentMandateMandateType[6]	PaymentMandate.MandateType[6]
PaymentMandateMandateStatus[6]	PaymentMandate.MandateStatus[6]
PaymentMandateAuthorizationType[6]	PaymentMandate.AuthorizationType[6]
PEFEFctrDataSourceType[4]	ProductEmissionsFactor.EmssnFctrDataSourceType[4]
PFATypeEnum[6]	PartyFinancialAsset.Type[6]
PFAStatusEnum[6]	PartyFinancialAsset.Status[6]
PFAOwnershipTypeEnum[6]	PartyFinancialAsset.OwnershipType[6]
PFAVerificationStatusEnum[6]	PartyFinancialAsset.VerificationStatus[6]
PersonEmploymentType[1]	PersonEmployment.EmploymentType[1]
PersonLanguageLanguage[1]	PersonLanguage.Language[1]

Standard Value Set Name (API version 38.0 and later)	Field Name (API version 37.0 and earlier)
PersonLanguageSpeakingProficiencyLevel[1]	PersonLanguage.SpeakingProficiencyLevel[1]
PersonLanguageWritingProficiencyLevel[1]	PersonLanguage.WritingProficiencyLevel[1]
PersonNameNameUsageType[1]	PersonName.NameUsageType[1]
PersonVerificationStatus[1]	PersonEmployment.VerificationStatus[1]
PHealthReactionSeverity[1]	PatientHealthReaction.Severity[1]
PIdentityVerificationResult	PartyIdentityVerification.Result
PIdentityVerificationStatus	PartyIdentityVerification.Status
PIVerificationStepStatus	PartyIdentityVerificationStep.Status
PIVerificationStepType	PartyIdentityVerificationStep.Type
PIVerificationVerifiedBy	PartyIdentityVerification.VerifiedBy
PIVOverriddenResult	PartyIdentityVerification.OverriddenResult
PIVResultOverrideReason	PartyIdentityVerificationResult.OverrideReason
PIVSVerificationDecision	PartyIdentityVerificationStep.VerificationDecision
PlaceOfService[1]	CareRequest.PlaceOfService[1]
PlanBenefitStatus[1]	PlanBenefit.Status[1]
PMDDosageDefinitionType[1]	PatientMedicationDosage.DosageDefinitionType[1]
PMDosageDosageAmountType[1]	PatientMedicationDosage.DosageQuantityType[1]
PMDosageRateType[1]	PatientMedicationDosage.DosageRateType[1]
PMPDetailDetailType[1]	PatientMedicalProcedureDetail.DetailType[1]
PMPOutcome[1]	PatientMedicalProcedure.Outcome[1]
PMPStatus[1]	PatientMedicalProcedure.Status[1]
PPCreditScoreProvider	PartyProfile.CreditScoreProvider
PPPrimaryIdentifierType	PartyProfile.PrimaryIdentifierType
PProfileAddressAddressType	PartyProfileAddress.AddressType
PProfileCountryOfDomicile	PartyProfile.CountryOfDomicile
PProfileEmploymentIndustry	PartyProfile.EmploymentIndustry
PProfileNationality	PartyProfile.Nationality
PProfileOffBoardingReason	PartyProfile.OffBoardingReason
PProfileRiskRiskCategory	PartyProfileRisk.RiskCategory
PPROverridenRiskCategory	PartyProfileRisk.OverridenRiskCategory

Standard Value Set Name (API version 38.0 and later)	Field Name (API version 37.0 and earlier)
PPTaxIdentificationType	PartyProfile.TaxpayerIdentificationType
ProblemCategory	Problem.Category
ProblemDefinitionCategory[1]	ProblemDefinition.Category[1]
ProblemDefinitionPriority[1]	ProblemDefinition.Priority[1]
ProblemDefinitionUsageTypeEnum[1]	ProblemDefinition.UsageType[1]
ProblemImpact	Problem.Impact
ProblemPriority	Problem.Priority
ProblemRelatedItemImpactLevel	ProblemRelatedItem.ImpactLevel
ProblemRelatedItemImpactType	ProblemRelatedItem.ImpactType
ProblemStatus	Problem.Status
ProblemSubCategory	Problem.SubCategory
ProblemUrgency	Problem.Urgency
ProcessExceptionCategory	ProcessException.Category
ProcessExceptionPriority	ProcessException.Priority
ProcessExceptionSeverity	ProcessException.Severity
ProcessExceptionStatus	ProcessException.Status
Product2Family	Product2.Family
ProductFeeFrequency[6]	ProductFee.Frequency[6]
ProductFeeType[6]	ProductFee.Type[6]
ProdRequestLineItemStatus	ProductRequestLineItem.Status
ProductLineEnum[1]	EngagementChannelType.UsageType[1]
ProductRequestStatus	ProductRequest.Status
ProgressionCriteriaMet[1]	CareRequest.CriteriaMet[1]
PScreeningStepResultCode	PartyScreeningStep.ResultCode
PScreeningStepStatus	PartyScreeningStep.Status
PSSResultOverrideReason	PartyScreeningSummary.ResultOverrideReason
PSStepMatchedFieldList	PartyScreeningStep.MatchedFieldList
PSSummaryScreenedBy	PartyScreeningSummaryS.creenedBy
PSSummaryScreeningDecision	PartyScreeningSummary.ScreeningDecision
PurchaserPlanAffiliation[1]	PurchaserPlan.Affiliation[1]

Standard Value Set Name (API version 38.0 and later)	Field Name (API version 37.0 and earlier)
PtyFinclLiabilityType[6]	PartyFinancialLiability.Type[6]
PtyFinclLiabilityShareType[6]	PartyFinancialLiability.ShareType[6]
PtyFinclLiabilityStatus[6]	PartyFinancialLiability.Status[6]
PtyFinclLiabilityVerficationStatus[6]	PartyFinancialLiability.VerficationStatus[6]
PurchaserPlanStatus[1]	PurchaserPlan.PlanStatus[1]
PurchaserPlanType[1]	PurchaserPlan.PlanType[1]
QuantityUnitOfMeasure	Product2.QuantityUnitOfMeasure ReturnOrderLineItem.QuantityUnitOfMeasure
QuestionOrigin[3]	Question.Origin[3]
QuickTextCategory	QuickText.Category
QuickTextChannel	QuickText.Channel
QuoteStatus	Quote.Status
ReceivedDocumentDirection[1]	ReceivedDocument.Direction[1]
ReceivedDocumentOcrStatus[1]	ReceivedDocument.OcrStatus[1]
ReceivedDocumentPriority[1]	ReceivedDocument.Priority[1]
ReceivedDocumentStatus[1]	ReceivedDocument.Status[1]
RegAuthCategory[1]	BusinessLicenseApplication.Category[1]
RegulatoryBodyType[1]	CareProviderAdverseAction.RegulatoryBodyType[1]
ReopenReason[1]	CareRequest.ReopenReason[1]
RequestedCareCodeType[1]	CareRequestItem.CodeType[1]
RequestedDrugCodeType[1]	CareRequestDrug.CodeType[1]
RequestedLevelOfCare[1]	CareRequest.RequestedLevelOfCare[1]
RequesterType[1]	CareRequest.RequesterType[1]
RequestingPractitionerLicense[1]	CareRequest.RequestingPractitionerLicense[1]
RequestingPractitionerSpecialty[1]	CareRequest.RequestingPractitionerSpecialty[1]
ResidenceStatusType[1]	BusinessLicense.ResidenceStatus[1]
RoleInTerritory2	UserTerritory2Association.RoleInTerritory2
ResourceAbsenceType	ResourceAbsence.Type
ReturnOrderLineItemProcessPlan	ReturnOrderLineItem.ProcessingPlan
ReturnOrderLineItemReasonForRejection	ReturnOrderLineItem.ReasonForRejection

Standard Value Set Name (API version 38.0 and later)	Field Name (API version 37.0 and earlier)
ReturnOrderLineItemReasonForReturn	ReturnOrderLineItem.ReasonForReturn
ReturnOrderLineItemRepaymentMethod	ReturnOrderLineItem.RepaymentMethod
ReturnOrderShipmentType	ReturnOrder.ShipmentType
ReturnOrderStatus	ReturnOrder.Status
RLAAppStatus	ResidentialLoanApplication.Status
RLAAmortType	ResidentialLoanApplication.AmortizationType
RLAAppliedFor	ResidentialLoanApplication.MortgageProgramType
RLAEstateType	ResidentialLoanApplication.EstateHoldType
RLALoanPurpose	ResidentialLoanApplication.LoanPurpose
RLAMortLienType	ResidentialLoanApplication.MortgageLienType
RLANativeTenure	ResidentialLoanApplication.NativeLandTenure
RLAProjectType	ResidentialLoanApplication.ProjectType
RLARefinanceType	ResidentialLoanApplication.RefinanceType
RLARefProgType	ResidentialLoanApplication.RefinanceProgramType
RLATitleType	ResidentialLoanApplication.TitleType
RLATrustType	ResidentialLoanApplication.TrustTitleType
SalesTeamRole	OpportunityTeamMember.TeamMemberRole UserAccountTeamMember.TeamMemberRole UserTeamMember.TeamMemberRole AccountTeamMember.TeamMemberRole
Salutation	CampaignMember.Salutation Contact.Salutation Lead.Salutation
SAppointmentGroupStatus[1]	ServiceAppointmentGroup.Status[1]
ScorecardMetricCategory	ScorecardMetric.Category
ScienceBasedTargetStatus[4]	SustainabilityScorecard.ScienceBasedTargetStatus[4]
SContributionSumCategory[4]	SocialContributionSum.Category[4]
Scope3CrbnFtprntStage[4]	Scope3CrbnFtprnt.FootprintStage[4]
ServiceAppointmentStatus	ServiceAppointment.Status
ServiceContractApprovalStatus	ServiceContract.ApprovalStatus

Standard Value Set Name (API version 38.0 and later)	Field Name (API version 37.0 and earlier)
ServicePlanTemplateStatus[1]	CarePlanTemplate.Status[1]
ServicingPractitionerLicense[1]	CareRequest.ServicingPractitionerLicense[1]
ServicingPractitionerSpecialty[1]	CareRequest.ServicingPractitionerSpecialty[1]
ServTerrMemRoleType	ServiceTerritoryMember.Role
ShiftStatus	Shift.Status
SocialContributionSumType[4]	SocialContributionSum.Type[4]
SocialPostClassification	SocialPost.Classification
SocialPostEngagementLevel	SocialPost.EngagementLevel
SocialPostReviewedStatus	SocialPost.ReviewedStatus
SolutionStatus	Solution.Status
SourceBusinessRegion[4]	Scope3EmssnSrc.BusinessRegion[4] StnryAssetEnvrSrc.BusinessRegion[4] VehicleAssetEmssnSrc.BusinessRegion[4]
StatusReason	Asset.StatusReason
StnryAssetCrbnFtprntStage[4]	StnryAssetCrbnFtprnt.FootprintStage[4]
StnryAstCrbnFtAllocStatus[4]	StnryAssetCrbnFtprnt.AllocationStatus[4]
StnryAstCrbnFtDataGapSts[4]	StnryAssetCrbnFtprnt.DataGapStatus[4]
StnryAstCrbnFtAllocStatus[4]	StnryAssetCrbnFtprnt.AllocationStatus[4]
StnryAstCrbnFtDataGapSts[4]	StnryAssetCrbnFtprnt.DataGapStatus[4]
StnryAstEvSrcStnryAstTyp[4]	StnryAssetEnvrSrc.StationaryAssetType[4]
StnryAssetWaterFtprntStage[4]	StnryAssetWaterFtprnt.FootprintStage[4]
SupplierClassification[4]	SustainabilityScorecard.SupplierClassification[4]
SupplierEmssnRdctnCmtTypev	SustainabilityScorecard.SupplierEmssnRdctnCmtType[4]
SupplierReportingScope[4]	SustainabilityScorecard.SupplierReportingScope[4]
SupplierTier[4]	SustainabilityScorecard.SupplierTier[4]
SustainabilityScorecardStatus[4]	SustainabilityScorecard.Status[4]
TaskPriority	Task.Priority
TaskStatus	Task.Status
TaskSubject	Task.Subject
TaskType	Task.Type

Standard Value Set Name (API version 38.0 and later)	Field Name (API version 37.0 and earlier)
TCDDetailType[1]	TrackedCommunicationDetail.DetailType[1]
TCPriority[1]	TrackedCommunication.Priority[1]
TCStatus[1]	TrackedCommunication.Status[1]
TCStatusReason[1]	TrackedCommunication.StatusReason[1]
TopicFailureReasonEnum[1]	EngagementTopic.ProcessFailureReason[1]
TopicProcessStatusEnum[1]	EngagementTopic.ProcessStatus[1]
TrackedCommunicationType[1]	TrackedCommunication.Type[1]
TypesOfIntervention[1]	CareInterventionType.InterventionType[1]
UnitOfMeasure	ConsumptionSchedule.UnitOfMeasure
UnitOfMeasureType[1]	UnitOfMeasure.Type[1]
VehicleAstCrbnFtprntStage[4]	VehicleAssetCrbnFtprnt.FootprintStage[4]
VehicleType[4]	VehicleAssetCrbnFtprnt.VehicleType[4] VehicleAssetEmssnSrc.VehicleType[4]
WasteFootprintStage[4]	WasteFootprint.FootprintStage[4]
WasteDisposalType[4]	WstDispoEmssnFctrSetItm.DisposalType[4]
WasteType[4]	WstDispoEmssnFctrSetItm.WasteType[4]
WorkOrderLineItemPriority	WorkOrderLineItem.Priority
WorkOrderLineItemStatus	WorkOrderLineItem.Status
WorkOrderPriority	WorkOrder.Priority
WorkOrderStatus	WorkOrder.Status
WorkStepStatus	WorkStep.Status
WorkTypeDefApptType	ServiceAppointment.AppointmentType
WorkTypeGroupAddInfo	WorkTypeGroup.AdditionalInformation

[1]Part of Salesforce Health Cloud.

[2]You can only update the label in this standard value set or picklist field. You can't insert or delete picklist values.

[3]You can't read or update this standard value set or picklist field.

[4]Part of Salesforce Net Zero Cloud.

[5]Part of Public Sector Solutions.

[6]Part of Digital Lending Solutions.

INDEX

C

ChatterEmailsMDSettings component 1869
Components
    ChatterEmailsMDSettings 1869

P

Prompt component 1639

U

UserEngagementSettings component 2132