

Version 4.0

Force.com Office Toolkit Developer's Guide

© Copyright 2000-2009 salesforce.com, inc. All rights reserved. Salesforce.com and the "no software" logo are registered trademarks, and AppExchange, "Success On Demand," and "The Business Web" are trademarks of salesforce.com, inc. All other trademarks mentioned in this document are the properties of their respective owners.

Table of Contents

ting Started	
Chapter 1: Introducing the Office Toolkit	9
Customize, Integrate, and Extend Your Salesforce Solutions	
Supported Salesforce Editions	
Development Platforms	10
Comparing the Office Toolkit and the API	
Office Toolkit Support Policy	11
Related Resources	
What's New in Version 4.0	
Quick Start	14
Chapter 2: Standard and Custom Object Basics	21
Primitive Data Types	
Field Types	23
API Data Types and Salesforce Field Types	
Core Data Types Used in API Calls	
System Fields	
Required Fields	
Frequently-Occurring Fields	
API Field Properties	
Relationships Among Objects	
Relabeling Fields and Tabs and the API	
Force.com AppExchange Object Prefixes and the API	
Custom Objects	
Chapter 3: API Call Basics	
Characteristics of API Calls	
Factors that Affect Data Access	
Synchronous and Asynchronous Calls	40
Single-Object and Batch Operations	40
Chapter 4: Error Handling	43
Error Handling for Session Expiration	44
More About Error Handling	44
Sample Error Handling—VBA	44
Chapter 5: Security and the API	45
User Authentication	46
User Profile Configuration	46
Security Token	46
Sharing	47

Table of Contents

	Implicit Restrictions for Objects and Fields	
	API Access in Force.com AppExchange Packages	
	Outbound Port Restrictions	
Refere	ence	51
(Chapter 6: Data Model	51
	- Sales Objects	
	Task and Event Objects	53
	Support Objects	54
	Document, Note, and Attachment Objects	54
	User and Profile Objects	
	Record Type Objects	56
	Product and Schedule Objects	
	Sharing and Team Selling Objects	
	Territory Management	
	Process Objects	
(Chapter 7: Standard Objects	60
	Account	
	AccountContactRole	75
	AccountHistory	
	AccountOwnerSharingRule	
	AccountPartner	
	AccountShare	
	AccountTag	
	AccountTeamMember	
	AccountTerritoryAssignmentRule	
	AccountTerritoryAssignmentRuleItem	
	AccountTerritorySharingRule	85
	ActivityHistory	
	AdditionalNumber	
	ApexClass	
	ApexComponent	
	ApexPage	94
	ApexTrigger	95
	Approval	
	Asset	
	AssetTag	
	AssignmentRule	
	AsyncApexJob	
	Attachment	
	Bookmark	
	BrandTemplate	
	BusinessHours	
	BusinessProcess	

CallCenter	111
Campaign	112
CampaignMember	117
CampaignMemberStatus	118
CampaignOwnerSharingRule	120
CampaignShare	120
Campaign Tag	121
Case	122
CaseComment	126
CaseContactRole	127
CaseHistory	128
CaseOwnerSharingRule	129
CaseShare	130
CaseSolution	131
CaseStatus	132
CaseTag	133
CaseTeamMember	134
CaseTeamRole	134
CaseTeamTemplate	135
CaseTeamTemplateMember	135
CaseTeamTemplateRecord	136
CategoryData	136
CategoryNode	137
CategoryNodeLocalization	138
Community	140
Contact	141
ContactHistory	145
ContactOwnerSharingRule	146
ContactShare	147
ContactTag	148
Contract	149
ContractContactRole	153
ContractHistory	154
ContractStatus	154
ContractTag	155
CurrencyType	156
DatedConversionRate	157
Division	158
Document	159
DocumentAttachmentMap	162
Document Tag	
EmailMessage	
EmailServicesAddress	165
EmailServicesFunction	166
EmailStatus	170

EmailTemplate	171
EntityHistory	
Event	
EventAttendee	
EventTag	
FiscalYearSettings	
Folder	
ForecastShare	
Group	
GroupMember	
Holiday	
Idea	
IdeaComment	
Lead	
LeadHistory	
LeadOwnerSharingRule	
LeadShare	
LeadStatus	
LeadTag	
LineitemOverride	
MailMergeTemplate	
Name	
Note	
NoteAndAttachment	
NoteTag	
OpenActivity	
Opportunity	
OpportunityCompetitor	
OpportunityContactRole	
OpportunityFieldHistory	
OpportunityHistory	
OpportunityLineItem	
OpportunityLineItemSchedule	
OpportunityOverride	
OpportunityOwnerSharingRule	
OpportunityPartner	
OpportunityShare	
OpportunityStage	
OpportunityTeamMember	
OpportunityTag	
Order	
OrderHistory	
OrderItem	
OrderTag	
Organization	
0	

Partner	
PartnerNetworkConnection	
PartnerNetworkRecordConnection	
PartnerRole	
Period	
Pricebook2	
PricebookEntry	
ProcessInstance	
ProcessInstanceHistory	
ProcessInstanceStep	
ProcessInstanceWorkitem	
Product2	
Profile	
QuantityForecast	
QuantityForecastHistory	
QueueSobject	
RecordType	
RevenueForecast	
RevenueForecastHistory	
Scontrol	
ScontrolLocalization	
SelfServiceUser	
Solution	
SolutionHistory	
SolutionStatus	
Solution Tag	
StaticResource	
TagDefinition	
Task	
TaskPriority	
TaskStatus	
TaskTag	
Territory	
User	
UserAccountTeamMember	
UserLicense	
UserPreference	
UserRole	
UserTeamMember	
UserTerritory	
Vote	
WebLink	
Weblink cooligation	324

ChildRelationship4	
Field4	
LocalizationContext4	
MergeRequest8	
PickListValue4	
QueryResultSet4	
SBriefcaseType	
SError	
SforceSession4	
SforceSessionEvents4	
SObject4	
Tab4	
TabSet4	
TabSetCollection4	
UserInfoResult8	
Chapter 9: Core Calls	
ConvertLead()	
Create()	
CreateObject()	
GetDeleted()	
GetUpdated()	
Login()	
Query()	
Refresh()	
Retrieve()	
Search()	
SetProxyInfo()	
SetServerURL()	
SetSOAPHeader()	
Undelete()	
Update()	
Chapter 10: Describe Calls	
DescribeGlobal()	
DescribeSObject()	
DescribeSObjects()	
describeSoftphoneLayout()	
DescribeTabs()	
Chapter 11: Utility Calls	436
GetServerTimestamp()	
GetUserInfo()	
ResetPassword()	
SetPassword()	

the API with Salesforce Features	443
Chapter 12: Implementation Considerations	443
Login Server URL	
Start By Logging In to the Login Server	
Typical API Call Sequence	
Multiple Instances of Salesforce.com Database Servers	
Monitoring API Traffic	
API Usage Metering	
Compression	
HTTP Persistent Connections	
HTTP Chunking	
Internationalization and Character Sets	
XML Compliance	
Office Toolkit and Proxies	
.Net, Non-String Fields, and the Enterprise WSDL	447
Chapter 13: Outbound Messaging	448
Understanding Outbound Messaging	
Understanding Notifications	
Setting Up Outbound Messaging	
Important Security Considerations	
Understanding the Outbound Messaging WSDL	
Building a Listener	454
Chapter 14: Data Loading and Integration	456
Client Application Design	
Salesforce Settings	
Best Practices with Any Data Loader	
Integration and Single Sign-On	459
Chapter 15: Data Replication	460
API Calls for Data Replication	
Scope of Data Replication	
Data Replication Steps	
Object-Specific Requirements for Data Replication	
Polling for Changes	
Checking for Structural Changes in the Object	463
Chapter 16: Feature-Specific Considerations	464
Archived Activities	
Person Account Record Types	
Opportunity Forecast Override Business Rules	
Call Centers and the API	
Implementing Salesforce Integrations on Force.com	

Table of Contents

[ndex	72

GETTING STARTED

Chapter 1

Introducing the Office Toolkit

In this chapter ...

- Customize, Integrate, and Extend Your Salesforce Solutions
- Supported Salesforce Editions
- Development Platforms
- Comparing the Office Toolkit and the API
- Office Toolkit Support Policy
- Related Resources
- What's New in Version 4.0
- Quick Start

Salesforce provides programmatic access to your organization's information using a simple, powerful, and secure application programming interface, the Force.com Web Services API (the API). To use this document, you should have a basic familiarity with software development, Web services, and the Salesforce user interface.

The Office Toolkit plug-in makes it easy for developers to access the API directly from within Microsoft Office products, simplifying the creation of new integrations and Office-based solutions.

Any functionality described in this guide is available only if your organization has the API feature enabled. This feature is enabled by default for Unlimited, Enterprise, and Developer Editions. Some Professional Edition organizations may also have the API enabled. If you cannot access the features you see in this guide, contact salesforce.com.



Note: Salesforce.com Education Services offers a suite of training courses to enable developers to design, create, integrate, and extend applications built on the Apex platform. Be sure to visit http://www.salesforce.com/training to learn more.

Customize, Integrate, and Extend Your Salesforce Solutions

The Force.com platform allows you to customize, integrate, and extend your Salesforce organization using the language and platform of your choice:

- **Customize Salesforce** with custom fields, links, objects, page layouts, buttons, record types, s-controls, and tabs to meet specific business requirements.
- Integrate Salesforce with your organization's ERP and finance systems, deliver real-time sales and support information to company portals, and populate critical business systems with customer information.
- Extend Salesforce in presentation, business logic, and data services with new functionality that reflects the business requirements of your organization.

For more information about Force.com solutions, developer resources, and community resources, go to Developer Force.

Supported Salesforce Editions

To use the API, your organization must use Enterprise Edition, Unlimited Edition, or Developer Edition. If you are an existing Salesforce customer and want to upgrade to either Enterprise or Unlimited Edition, contact your account representative.

To develop Web service client applications, it is strongly recommended that you use Developer Sandbox, which is an exact replica of your Salesforce deployment, including all customization and data. For more information, see http://www.salesforce.com/products/sandbox.jsp.

Developer Edition provides access to all of the features available with Enterprise Edition. Developer Edition is constrained only by the number of users and the amount of storage space. Developer Edition provides a development context that allows you to build and test your solutions without affecting your organization's live data. Developer Edition accounts are available for free at http://wiki.apexdevnet.com/index.php/Getting_Started.

Development Platforms

The Office Toolkit supports all COM-aware development environments, including Visual Basic 6.0, VBA, C++, .NET, JavaScript (in Microsoft Internet Explorer), VBScript, and others.



Note: Throughout this document, code examples written in VBA are used to demonstrate coding practices for COM client applications that use the API. However, your client application may use a different language—such as those just described—in your COM-aware development environment.

Comparing the Office Toolkit and the API

The Office Toolkit version of the API provides a Component Object Model (COM) interface to developers who want to write client applications that use COM technology to access their organization's Salesforce data. The Office Toolkit is built on top of, and interacts with, an organization's Salesforce data via the API. The Office Toolkit acts as an intermediary between COM client applications and the API, handling certain tasks implicitly so that client applications are simpler to code.

The Office Toolkit version of the API provides much of the same functionality that the API provides, with significant differences in the way client applications implement API calls. For readers who are familiar with the API, the following list provides a summary of key similarities and differences:

- The Office Toolkit supports the same Salesforce data model as the API (see Standard Objects and Data Model). In the API, objects are uniquely typed (an account is represented by an Account object), whereas in the Office Toolkit, all objects are instances of an SObject4 (in which the ObjectType property in the SObject4 specifies the type of object it is (ObjectType="Account").
- The Office Toolkit supports many of the same API calls, although the names of API calls are capitalized in the Office Toolkit (Login() instead of login), and the call syntax is different.
- The API and the Office Toolkit both support versions of the following calls: Login(); Create() and Update(); Retrieve(); Query() and the Salesforce Object Query Language (SOQL) (although the Office Toolkit provides additional wildcard support in the *fieldExpression*); Search() and the Salesforce Object Search Language (SOSL); and GetDeleted() and GetUpdated() (Data Replication).
- The Office Toolkit provides calls for operations that are handled differently in the API, such as setting the server URL (SetServerURL()) and setting the SOAP header (SetSOAPHeader()).
- The Office Toolkit does not expose the API call queryMore because this operation is handled implicitly when a client application traverses the result set of a Query() call.
- The Office Toolkit does not expose the getUserInfo call of the API because this information is automatically returned in the SforceSession4 object following a successful Login().
- The Office Toolkit does not expose the getServerTimestamp call of the API because a client application can obtain this information by reading the CurrentServerTime property in the SforceSession4 object.
- The Office Toolkit does not expose the describeGlobal call of the API because a client application can obtain the list of available objects by traversing the EntityNames property in the SforceSession4 object.
- The Office Toolkit does not expose the describeSObject call of the API because a client application can inspect the properties of an SObject4 for a given ObjectType .
- The Office Toolkit does not support inspecting layouts and tabs via the API (describeLayout and describeTabs calls, respectively, in the API) or changing passwords (setPassword and resetPassword calls).
- The Office Toolkit supports both synchronous and asynchronous calls (see Synchronous and Asynchronous Calls). The API supports only synchronous calls.
- Both the API and the Office Toolkit support operations on single objects (such as creating a single Account) as well as batch operations (such as updating a group of Contact records), although the programming techniques differ.
- Operations in the API and Office Toolkit are governed by the same factors that affect data access (see Factors that Affect Data Access).
- Both the API and the Office Toolkit support many of the same error codes (see SError), although error handling techniques vary among languages used to develop client applications.
- The Office Toolkit represents data in objects as COM data types, as described in Field Types. Note that, for Base64 type data (such as Attachment records), client applications are responsible for the conversion of Base64 data between binary and String formats.

There are other minor differences. When in doubt, consult the relevant section in this document. The rest of this document includes the Office Toolkit when it references the API. For more information about the API exclusive of the Office Toolkit, see the *Force.com Web Services Developer's Guide*.

Office Toolkit Support Policy

When a new version of the Office Toolkit is released, it becomes the supported version, and earlier releases are not supported.

Related Resources

The salesforce.com 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://wiki.apexdevnet.com/index.php/Getting_Started for more information, or visit https://developer.force.com/join to sign up for a free Developer Edition account.

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

- Salesforce.com for information about the Salesforce application.
- Force.com AppExchange for access to apps created for Salesforce.
- Salesforce.com Community for services to ensure Salesforce customer success.

What's New in Version 4.0

Generally Available Enhancements

The Force.com Office Toolkit has been improved for Spring '09:

New Objects

The following new objects have been added to the Office Toolkit version 4.0:

- MergeRequest8: Represents a structure that is passed as a parameter to merge records of the same object type into one of the records.
- SforceSessionEvents4: Events raised by the SforceSession4 object
- Tab4: Provides information about the standard and custom tabs available to the logged-in user.
- TabSet4: Provides information about the standard and custom apps available to the logged-in user. An app is a group of tabs that works as a unit to provide application functionality. For example, two of the standard Salesforce apps are "Sales" and "Service and Support."
- TabSetCollection4: This object is a collection of the standard and custom apps in your organization. You can use the collection to get information about the apps available to the logged-in user.
- UserInfoResult8: Contains the personal information about the currently logged-in user. This information includes common profile data that your client application can use for display purposes, performing currency calculations, and so on.

Changed Objects

The following COM objects have changed in the Office Toolkit version 4.0:

- ChildRelationship3 This object is now named ChildRelationship4.
- Field3 -This object is now named Field4.It contains the following new properties and functions:
 - CalculatedFormula
 - CaseSensitive
 - ControllerName
 - DefaultValueFormula
 - DependentFields
 - ExternalId

- GetDependentPicklistValues
- HtmlFormatted
- IsDependentPicklist
- IsMasterPicklist
- NamePointing
- RelationshipName
- Sortable
- Unique
- LocalizationContext3 This object is now named LocalizationContext4.
- PicklistValue3 This object is now named PicklistValue4. It contains the following new property: ValidFor.
- QueryResultSet3 This object is now named QueryResultSet4. It contains the following new properties:
 - LastConfigChange
 - Removed
 - Token
- SforceSession3 This object is now named SforceSession4.

It includes the following new properties and functions:

- CurrentUserInfo
- DoLookupSearch
- DoTZConversions
- EarliestStartDateAvailable
- GetOfflineData
- GetTabSets
- LatestDateAvailable
- MakeHTTPRequest
- Mobile
- QueryAll
- OnAckOfflineDataFinished
- OnGetOfflineDataFinished
- Undelete
- UseCache
- AckOfflineData
- DescribeSObjects
- SObject3 This object is now named is now named SObject4. It includes the following new properties and functions:
 - GetJoinResults
 - Mergeable
 - Retrievable

New Calls

The following calls are now supported in Office Toolkit version 4.0.

- DescribeGlobal()
- DescribeSObject()
- DescribeSObjects()

- DescribeTabs()
- GetServerTimestamp()
- GetUserInfo()
- ResetPassword()
- SetPassword()

Removed Calls

The following calls are no longer supported in the Office Toolkit version 4.0.

- Delete()
- EmptyRecycleBin()
- InvalidateSessions()
- Logout()
- Merge()
- Process()
- QueryAll()

Quick Start

Before you can use the API, your user profile must have the "API Enabled" permission selected. This permission is enabled by default. For more information see the help in the user interface. Use this topic to create a sample application in your development environment.



Note: Before you begin building integrations or client applications, install your development platform according to its product documentation. In addition, you should read the rest of this document.

Step 1: Obtain a Salesforce Developer Edition Account

If you are not already a member of the developer community, go to http://developer.force.com/join and follow the instructions for signing up for a Developer Edition account. Even if you already have an Enterprise Edition or Unlimited Edition account, it is strongly recommended that you use Developer Edition for developing, staging, and testing your solutions against sample data to protect your organization's live data. This is especially true for applications that will be inserting, updating, or deleting data (as opposed to simply reading data).

Step 2: Obtain the Office Toolkit DLL

Go to the Office Toolkit page at http://www.salesforce.com/developer. Follow the instructions for downloading and installing the Office Toolkit DLL.

Step 3: Import the Office Toolkit DLL Into Your Development Environment

Import the Office Toolkit DLL (SForceOfficeToolkitLib3.dll) into your development environment. For example, if you are using Visual Basic 6 to develop your COM client application, then follow these instructions to import the Office Toolkit DLL into your project:

- 1. Open your project in Microsoft Visual Basic 6.
- 2. Choose Tools | References | Office Toolkit 4.0.

- 3. To verify that the Office Toolkit DLL is working, from the View menu, choose Object Browser (or just press F2).
- 4. From the Libraries list, click SForceOfficeToolkitLib3.
- 5. Navigate the library to view its contents.

Step 4: Walk Through the Sample Code

Once you have imported your DLL library, you can begin building client applications that use the API. The following example VBA client application uses the API to interact with Salesforce data. The modules in this section are all called from the RunSampleCode subroutine, except the last example:

- Declaring an Object
- Logging in to the API
- Creating a New Account
- Creating a New Account Using Single-Object Operations
- Creating a New Account Using Batch Operations
- Updating the Account
- Running a Query
- Examining Fields in an SObject4
- Updating an Array of SObject4 by Id

Declaring an Object

The following code declares the Office Toolkit object.

```
Option Explicit
'Declare an OfficeToolkit object.
'The WithEvents keyword enables asynchronous function calls
Dim WithEvents g sfApi As SForceOfficeToolkitLib3.SForceSession3
```

Logging in to the API

The following code creates an instance of the SforceSession4 object, calls Login () to log in to the API, and upon successful login, obtains the populated SforceSession4 object returned from the API.

```
Public Function SampleLogin(UserName, Password) As Boolean
On Error Resume Next
'create a session object
Set g_sfApi = New SForceOfficeToolkitLib4.SForceSession4
'make a login call
SampleLogin = g_sfApi.Login(UserName, Password)
End Function
```

Creating a New Account

The following code creates a single account in two different ways:

- Single-object operations work on single object instances.
- Batch operations that work on batches of objects (one or more objects).

For more information, see Single-Object and Batch Operations.

Creating a New Account Using Single-Object Operations

The following section demonstrates the use of single-object operations. The client application creates a new SObject4 object by calling CreateObject(), populates the fields of the SObject4 (specifying an account name), and calls the Create function on that account to commit the changes in the organization's Salesforce data. Thereafter, it shows calling the Refresh function on the SObject4 to refresh the local SObject4 instance with the latest Salesforce data.

```
'Create a blank object
Public Function CreateAccount() As SObject4
On Error GoTo handleError
'all entities are of type SObject4
Dim account As SObject4
'create a blank object
Set account = g_sfApi.CreateObject("account")
account("name") = "Sample"
'commit the object
account.Create
'refresh the object to get server set values
account.Refresh
'set as the function return value
Set CreateAccount = account
```

Creating a New Account Using Batch Operations

The following section demonstrates the use of batch operations. The client application dimensions an SObject4 array, creates the first element in the array as a new SObject4 object by calling CreateObject(), populates the first element in the array with data (specifying the account name), and calls Create() to commit the changes in the organization's Salesforce data. Thereafter, it shows calling Refresh() to refresh the local SObject4 instance with the latest Salesforce data.

```
'create using batch call
Dim so(0) As SObject4
Set so(0) = g_sfApi.CreateObject("account")
so(0)("name") = "Test Array account"
'call batch method
g_sfApi.Create so, False
'call batch refresh
```

g_sfApi.Refresh so, False Exit Function

The following section handles any errors.

```
handleError:
MsgBox g_sfApi.ErrorMessage
End Function
```

Updating the Account

Now that the Account has been added, this module shows how to update the account using single-object and batch operations:

- It populates the SObject4 with data and then calls the Update function on the SObject4 to commit changes to the Salesforce data.
- It adds the SObject4 to an SObject4 array and then calls Update () on the SforceSession4 object to commit changes to the Salesforce data.

```
Function SyncUpdate(account As SObject4)
On Error GoTo handleError
Dim so(0) As SObject4
account("name").value = "New Value"
'you can call Update on the SObject (single-object operation, synchronous)
account.Update
'alternatively, you can call Update on the session (batch operation)
'first, put the account into an array
Set so(0) = account
'call synchronous batch update
g_sfApi.Update so, False
Exit Function
handleError:
MsgBox g_sfApi.ErrorMessage
End Function
```

Running a Query

This module shows how to run a query and examine the query results. It dimensions an array of QueryResultSet4 objects, submits the query to the API via the Update() call on the SforceSession4 object, and navigates the returned QueryResultSet4 array that contains the query results.

```
Public Function Query()
```

```
Dim qr As QueryResultSet4
Dim v As Variant
Dim s As SObject4
Set qr = g_sfApi.Query("select * from task", False)
For Each v In qr
'loop through the results
'cast to an SObject4 to see more helpful debug info
Set s = v
'use the object
    s("Name") = "Query"
    's.Update
Next v
End Function
```

Examining Fields in an SObject4

This final module shows how to check fields and properties in an SObject4 object, as well as how to examine picklist values.

```
'Check all the fields in the sobject, look at some attributes
Public Function CheckFields (account As SObject4)
 Dim v, plv As Variant
 Dim i, x As Integer
 Dim f As Field4
 Dim pl As PickListValue4
'use for each to look at the fields
  For Each v In account.Fields
'cast to a field to get better info
 Set f = v
 Debug.Print f.Name
 Debug.Print f.AutoNumber
 Debug.Print f.ByteLength
 Debug.Print f.Createable
 Debug.Print f.Custom
 Debug.Print f.DefaultOnCreate
 Debug.Print f.Digits
 Debug.Print f.Filterable
 Debug.Print f.IsCalculated
 Debug.Print f.Label
 Debug.Print f.Length
 Debug.Print f.NameField
 Debug.Print f.Nillable
 Debug.Print f.Precision
 Debug.Print f.ReferenceTo
 Debug.Print f.RestrictedPicklist
 Debug.Print f.Scale
Debug.Print f.Type
```

```
Debug.Print f.Updateable
 Debug.Print f.VariantType
'look at picklist values
 If f.Type = "picklist" Then
 For Each plv In f.PickListValues
 Set pl = plv
 Debug.Print pl.IsDefault
 Debug.Print pl.Label
 Debug.Print pl.Active
 Debug.Print pl.value
Next plv
 End If
Next v
'use an array index to look at the fields
'since the reference is strongly typed, you get more debug info
For i = 0 To UBound (account.Fields)
Debug.Print account.Fields(i).Name
Next i
End Function
```

Updating an Array of SObject4 by Id

This example shows how to update an array of SObject4 by the object Id.

```
Function UpdateArray()
```

```
On Error GoTo handleError
'create array of sobjects
Dim so(2) As SObject4
'only need to set the id
Set so(0) = g_sfApi.CreateObject("opportunity")
so(0)("name") = "new name"
so(0)("id") = "0060000007BONP"
Set so(1) = g_sfApi.CreateObject("opportunity")
so(1)("name") = "new name 2"
so(1)("id") = "0060000007ENIL"
'call synchronous batch update
g_sfApi.Update so, False
Exit Function
```

handleError:
'session level error
MsgBox g_sfApi.ErrorMessage
'which object was in error?
CheckArrayError g_sfApi.Error, so
End Function

Chapter 2

Standard and Custom Object Basics

In this chapter ...

- Primitive Data Types
- Field Types
- API Data Types and Salesforce Field Types
- Core Data Types Used in API Calls
- System Fields
- Required Fields
- Frequently-Occurring Fields
- API Field Properties
- Relationships Among Objects
- Relabeling Fields and Tabs and the API
- Force.com AppExchange Object Prefixes and the API
- Custom Objects

Generally speaking, API objects represent database tables that contain your organization's information. For example, the central object in the Salesforce data model represents accounts—companies and organizations involved with your business, such as customers, partners, and competitors. The term "record" describes a particular occurrence of an object (such as a specific account like "IBM" or "United Airlines" that is represented by an Account object). A record is analogous to a row in a database table. In the Office Toolkit, objects are represented by SObject4 objects, and fields within the object are represented by Field4 objects.

Objects already created for you by Salesforce are called standard objects. Objects you create in your org are called custom objects. For a list of standard objects, see the Standard Objects.

While this document describes all of the objects available in the API, your applications work with only the objects that you are authorized to access. Programmatic access to objects is determined by the objects that available in the EntityNames [] array in the SforceSession4 object, your organization configuration, your security access (which is configured by your organization's system administrator in your personal profile), and your data sharing model. For more information, see Factors that Affect Data Access.

Most of the objects accessible through the API are read-write objects. However, there are a few objects that are read-only. This fact is noted in the description for the object.



Note: The objects Pricebook and Product are deprecated, and have been removed from this document.

Primitive Data Types

The API uses the following primitive data types:

Value	Description
base64	Base 64-encoded binary data. Fields of this type are used for storing binary files in Attachment records, Document records, and Scontrol records. In these objects, the Body or Binary field contains the (base64 encoded) data, while the BodyLength field defines the length of the data in the Body or Binary field. In the Document object, you can specify a URL to the document instead of storing the document directly in the record.
	Note: Client applications are responsible for the conversion of Base64 data between binary and string formats.
boolean	Boolean fields have one of these values: true (or 1), or false (or 0).
byte	A set of bits.
date	Date data. Fields of this type contain date values, such as ActivityDate in the Event object. Unlike dateTime fields, date fields contain no time value—the time portion of a date field is not relevant and is always set to midnight in the Coordinated Universal Time (UTC) time zone.
	If you specify a date value in a query, you can filter on date fields only.
dateTime	Date/time values (timestamps). Fields of this type handle date/time values (timestamps), such as ActivityDateTime in the Event object or the CreatedDate, LastModifiedDate, or SystemModstamp in many objects. Regular dateTime fields are full timestamps with a precision of one second. They are always transferred in the Coordinated Universal Time (UTC) time zone. The Office Toolkit automatically converts the Coordinated Universal Time (UTC) time zone into the local time zone of the client application, except for queries, GetUpdated(), and GetDeleted(). For these, the values are passed to the API without conversion.
	If you specify a dateTime value in a query, you can filter on dateTime fields only.
	Development tools differ in the way that they handle time data. Some development tools report the local time, while others report only the Coordinated Universal Time (UTC) time zone. To determine how your development tool handles time values, refer to its documentation.
	Note: The Event object has a DurationInMinutes field that specifies the number of minutes for an event. Even though this is a temporal value, it is an integer type—not a dateTime type.
double	Double values. Fields of this type can contain fractional portions (digits to the right of the decimal place), such as ConversionRate in CurrencyType. In the API, all non-integer values (such as Currency Field Type and Percent Field Type) contain values of type double. Some restrictions may be applied to double values:
	 Scale: Maximum number of digits to the right of the decimal place. Precision: Total number of digits, including those to the left and the right of the decimal place

Value	Description
	The maximum number of digits to the left of the decimal place is equal to Precision minus Scale. In the online application, precision is defined differently—it is the maximum number of digits allowed to the left of the decimal place.
	Values can be stored in scientific notation if the number is large enough (or, for negative numbers, small enough), as indicated by the W3C XML Schema Part 2: Datatypes Second Edition specification.
int	Fields of this type contain numbers with no fractional portion (digits to the right of a decimal place), such as the NumberOfEmployees in an Account. For integer fields, the digits field specifies the maximum number of digits that an integer can have.
string	Character strings. Fields that are of data type string contain text and some have length restrictions depending on the data being stored. For example, in the Contact object, the FirstName field is 40 characters, the LastName field is 80 characters, the MailingStreet is 255 characters.
	Note: For fields that contain strings, behavior is different beginning with API version 15.0. In API versions previous to 15.0, if you specify a value for a field, and that value is too large, the value is truncated. For API version 15.0 and later, if a value is specified that is too large, the operation fails and the fault code STRING_TOO_LONG is returned.
time	Time values. Fields of this type handle time values, such as FridayEndTime in the BusinessHours object.
	Development tools differ in the way that they handle time data. Some development tools report the local time, while others report only the Coordinated Universal Time (UTC) time zone. To determine how your development tool handles time values, refer to its documentation.

These data types are used in the SOAP 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 SOAP data types.

Primitive types are used as a standardized way to define, send, receive, and interpret basic data types in the SOAP messages exchanged between client applications and the API. In addition, primitive data types are interpreted in a Salesforce-specific way, which is useful for display formatting and for numeric conversion (adding values of different currencies).

For example, Salesforce chooses to interpret a double value passed via SOAP as a double in a number of possible ways, depending on the field definition. If the field type for that data is currency, Salesforce handles the display of the data by prepending it with a currency symbol and inserting a decimal for precision. Similarly, if the field type is percent, Salesforce handles the display of the data by appending a percent sign (%). Regardless of the field type, however, the value is sent in the SOAP message as a double.

The API uses data types called field types that are defined in the WSDLs. For more information, see Field Types.

For a list of COM data types associated with field types in objects, see Field Types.

Field Types

In addition to the primitive data types explained in Primitive Data Types, the API defines the following data types for object fields (which correspond to the list of possible values in the VariantType property in the Field4 object):



Note: For fields that contain strings, behavior is different beginning with API version 15.0. In API versions previous to 15.0, if you specify a value for a field, and that value is too large, the value is truncated. For API version 15.0 and later, if a value is specified that is too large, the operation fails and the fault code STRING TOO LONG is returned.

Field Type	What the Field Contains
anyType	Polymorphic data type that returns string, picklist, reference, Boolean, currency, int, double, percent, ID, date, datetime, url, or email data depending on the kind of field involved.
calculated	Fields that are defined by a formula. See Calculated Field Type.
comobobox	A comobobox, which includes a set of enumerated values and allows the user to specify a value not in the list. See comobobox Field Type.
currency	Currency values. See Currency Field Type.
email	Email addresses. See Email Field Type.
encryptedstring	Encrypted text fields contain any combination of letters, numbers, or symbols that are stored in encrypted form. You can set a maximum length of up to 175 characters. Available in API versions 11.0 and later.
ID	Primary key field for the object. See ID Field Type.
masterrecord	When records are merged, the ID of the record that is saved (the other records are deleted).
multipicklist	Multi-select picklists, which include a set of enumerated values from which multiple values can be selected. See Mult-Select Picklist Field Type.
percent	Percentage values. See Percent Field Type.
phone	Phone numbers. Values can include alphabetic characters. Client applications are responsible for phone number formatting. See Phone Field Type.
picklist	Picklists, which include a set of enumerated values from which one value can be selected. See Picklist Field Type.
reference	Cross-references to a different object. Analogous to a foreign key field in SQL. See Reference Field Type.
textarea	String that is displayed as a multiline text field. See Textarea Field Type.
url	URL values. Client applications should commonly display these as hyperlinks. See URL Field Type.

These field types extend the primitive data types, which are described in Primitive Data Types. While many of these field types follow common data typing conventions that are made explicit in their metadata, certain field types have unique characteristics that you need to understand before using them in your client application.

These field types apply to both standard and custom fields. They are used in the Type property inField4.

The following table shows the mapping between each API field type and the associated data type in COM applications.

API Field Type	Associated COM Type
base64	VT_BSTR
boolean	VT_BOOL
calculated	VT_BSTR

API Field Type	Associated COM Type
comobobox	VT_BSTR
currency	VT_DECIMAL
date	VT_DATE
datetime	VT_DATE
double	VT_DECIMAL
email	VT_BSTR
id	VT_BSTR
int	VT_I4
multipicklist	VT_BSTR
percent	VT_DECIMAL
phone	VT_BSTR
picklist	VT_BSTR
reference	VT_BSTR
string	VT_BSTR
textarea	VT_BSTR
url	VT_BSTR



Note: Some numeric fields have precision and scale limits. In addition, certain text fields have length restrictions. These restrictions are enforced when you Create() or Update() objects. However, the API may return data that does not meet these restrictions.

Null values will always be VT_EMPTY in any field, whether the value is programmatically set or is read from a server response.

Calculated Field Type

Calculated fields are read-only fields in the API. These are fields defined by a formula, which is an algorithm that derives its value from other fields, expressions, or values. You can filter on these fields in SOQL, but you should not replicate these fields. The length of text calculated fields is 3900 characters or less—anything longer will be truncated.

Calculated fields are called formula fields in the Salesforce user interface. For more information, see the online help topic Building Formulas.

comobobox Field Type

A comobobox is a picklist that also allows users to type a value that is not already specified in the list. A comobobox is defined as a string value.

Currency Field Type

Currency fields contain currency values, such as the ExpectedRevenue field in a Campaign, and are defined as type double.

For organizations that have the multicurrency option enabled, the CurrencyIsoCode field is defined for any object that can have currency fields. The CurrencyIsoCode field and currency fields are linked in a special way. On any specific record, the

CurrencyIsoCode field defines the currency of that record, and thus, the values of all currency fields on that record will be expressed in that currency.

For most cases, clients do not need to consider the linking of the CurrencyIsoCode field and the currency fields on an object. However, clients may need to consider the following:

- The CurrencyIsoCode field exists only for those organizations that have enabled multicurrency support.
- When displaying the currency values in a user interface, it is preferred to prepend each currency value with its CurrencyIsoCode value and a space separator.
- The CurrencyIsoCode field is a restricted picklist field. The set of allowable values, defined in the CurrencyType object, can vary from organization to organization. Attempting to set it to a value that is not defined for an organization causes the operation to be rejected.
- If you update the CurrencyIsoCode field on an object, it implicitly converts all currency values on that object to the new currency code, using the conversion rates that are defined for that organization in the Salesforce user interface. If you specify currency values in that same Update() call, the new currency values you specify are interpreted in the new CurrencyIsoCode field value, without conversion.
- The picklist values in a CurrencyIsoCode field do not exactly match the labels displayed in Salesforce.

To perform currency conversions, client applications can look up the CurrencyIsoCode in the CurrencyType object.

Email Field Type

Email fields contain email addresses. Client applications are responsible for specifying valid and properly formatted email addresses in Create() and Update() calls.



Note: Note that client applications are responsible for the conversion of Base64 data between binary and String formats.

ID Field Type

With rare exceptions, all objects in the API have a field of type ID that is named Id and contains a unique identifier for each record in the object. It is analogous a primary key in relational databases. When you Create() a new record, the Web service generates an ID value for the record, ensuring that it is properly formatted and unique within your organization's data. You cannot use the Update() call on ID fields. Because the ID value stays constant over the lifetime of the record, you can refer to the record by its ID value in subsequent API calls. Also, the ID value contains a three-character code that identifies the object type, which client applications can inspect in the returned SObject4 for a given record.

In addition, certain objects, including custom objects, have one or more fields of type reference that contain the ID value for a related record. These fields have names that end in the suffix "-Id", for example, <code>OwnerId</code> in the account object. <code>OwnerId</code> contains the ID of the user who owns that account. Unlike the field named <code>Id</code>, <code>reference</code> fields are analogous to foreign keys and can be changed via the <code>Update()</code> call. For more information, See Reference Field Type.

Some API calls, such as Retrieve(), accept an array of IDs as parameters—each array element uniquely identifies the row to retrieve or delete. Similarly, the Update() call accepts an array of SObject4 recordss—each SObject4 includes a Field4 object (in which the name="Id") that uniquely identifies the SObject4.

ID fields in the Salesforce user interface contain 15-character, base-62, case-sensitive strings. Each of the 15 characters can be a numeric digit (0-9), a lowercase letter (a-z), or an uppercase letter (A-Z). Two unique IDs may only be different by a change in case.

Because there are applications like Access which do not recognize that 50130000000014c is a different ID from 5013000000014C, an 18-digit, case-safe version of the ID is returned by all API calls. The 18 character IDs have been formed by adding a suffix to each ID in the Force.com API. 18-character IDs can be safely compared for uniqueness by case-insensitive applications, and can be used in all API calls when creating, editing, or deleting data.

If you need to convert the 18-character ID to a 15-character version, truncate the last three characters. Salesforce.com recommends that you use the 18-character ID.

Mult-Select Picklist Field Type

Multi-select picklist fields contain a list of one or more items from which a user can choose multiple items. One of the items can be configured as the default item. Selections are maintained as a string containing a series of attributes delimited by semicolons. For example, a query might return the values of a multivalue picklist as "first value; second value; third value". For information on querying multi-select picklists, See Querying Multi-Select Picklists.

Percent Field Type

Percent fields contain percent values. Percent fields are defined as type double.

Phone Field Type

Phone fields contain phone numbers, which can include alphabetic characters. Client applications are responsible for phone number formatting.

Picklist Field Type

Picklist fields contain a list of one or more items from which a user chooses a single item. They display as drop-down lists in the Salesforce user interface. One of the items can be configured as the default item.

In the Field4 object, the RestrictedPicklist property defines whether the field is a restricted picklist or not. The API does not enforce the list of values for advisory (unrestricted) picklist fields on Create() or Update(). When inserting an unrestricted picklist field that does not have a PickListValue4, the system creates an "inactive" picklist value. This value can be promoted to an "active" picklist value by adding the picklist value in the Salesforce user interface.

When creating new, inactive picklists, the API checks to see if there is a match. This check is case-insensitive.

In the Field4 object, the field PickListValues property contains an array of items (PickListValue4 objects). Each PickListValue4 defines the item's label, value, and whether it is the default item in the picklist (a picklist has no more than one default value).

Enumerated fields support localization of the labels to the language of the user. For example, for the Industry field on an Account, the value "Agriculture" may be translated to various languages. The enumerated field values are fixed and do not change with a user's language. However, each value may have a specified "label" field that provides the localized label for that value. You must always use the value when inserting or updating a field. The Query () call always returns the value, not the label. The corresponding label for a value should be used when displaying the value to the user in any user interface.

The API supports the retrieval of the certain picklists in the following objects: CaseStatus, ContractStatus, LeadStatus, OpportunityStage, PartnerRole, SolutionStatus, TaskPriority, and TaskStatus. Each object represents a value in the respective picklist. These picklist entries always specify some other piece of information, such as whether the status is converted, and so on. Your client application can invoke the Query() call on any of these objects (such as CaseStatus) to retrieve the set of values in the picklist, and then use that information while processing other objects (such as Case objects) to find more information about those objects (such as a given case). These objects are read-only via the API. To modify items in picklists, you must use the Salesforce user interface.

Reference Field Type

A reference field contains an Id value that points to a unique record (usually the parent record) on another object. This is analogous to the concept of a foreign key in relational databases. The name of a reference field ends, by convention, with the letters Id (such as CaseId orOpportunityId). For example, in the OpportunityCompetitor object, the OpportunityId field is a reference field that points to the Opportunity object. It contains an ID value that uniquely identifies an Opportunity record.

In some cases, an object can refer to another object of its same type. For example, an Account can have a parent link that points to another Account.

The Event and Task objects both have WhoId and WhatId cross-reference ID fields. Each of these cross-reference fields can point to one of several other objects. The WhoId field can point to a Contact or Lead, and the WhatId field can point to an Account, Opportunity, Campaign, or Case. In addition, if the WhoId field refers to a Lead, then the WhatId field must be empty.

You can describe and query each cross-referenced object. When you query a cross-reference ID field, it returns an object ID of the appropriate type. You can then query that ID to get additional information about the object, using the ID in the id field for that query.

The cross-reference ID field value is either:

- a valid record in your organization, or
- an empty value, which indicates an empty reference

The cross-reference ID field value, if non-null, is guaranteed to be an object in your organization. However, it is not guaranteed that you can query that object. Users with the "View All Data" permission can always query that object. Other users may be restricted from viewing or editing the referenced object.

When specifying a value for a cross-reference ID field in a Create() or Update() call, the value must be a valid Id value, and the user must have appropriate access to that object. The exact requirements vary from field to field.

For information on IDs, see ID Field Type.

Textarea Field Type

Textarea fields contain text that can be longer than 4000 bytes. Unlike string fields, textarea fields cannot be specified in the WHERE clause of a queryString of a Query() call. To filter records on this field, you must do so while processing records in the QueryResultSet4. For fields with this restriction, its Filterable property in the Field4 object is false.

URL Field Type

URL fields contain URLs. Client applications are responsible for specifying valid and properly formatted URLs in Create() and Update() calls.

API Data Types and Salesforce Field Types

Generally, API data types and Salesforce field types have the same names. For example, a date field is represented by a date data type in the API. However, some field types are represented differently depending on whether you are inspecting an object via the API or the Salesforce user interface. The following table contains the mapping for field types and data types that are different:

API Data Type	Corresponding Field Types in the Salesforce User Interface
ID	Lookup relationship, master-detail relationship
string	Auto number, email, phone, picklist, multi-select picklist, text, text area, long text area, and URL. Different maximum lengths are specified in the WSDL for text, text area, and long text area.
boolean	Checkbox
double	Currency, formula, number, percent, and roll-up summary
Varies by type	When formula fields are created in the Salesforce user interface, a type must be specified. This type corresponds to the API data type of the same name: currency, date, date/time, number, percent, or text.

All other fields that you can create in the Salesforce user interface fall into one of the following categories:

- The field is not available in both the Salesforce user interface and the API. For example, the BusinessHours object has fields of API data type time, but you cannot create a custom field of this type.
- Field types are the same as their corresponding API data type. For example, if you create a date field in the Salesforce user interface, that field is the date data type in the API.

For more information about API data types, see Primitive Data Types and Field Types.

Core Data Types Used in API Calls

Many calls in the API use the following data types:

- ID (String). See ID Field Type.
- ChildRelationship4
- LocalizationContext4
- MergeRequest8
- Field4
- QueryResultSet4
- PickListValue4
- SBriefcaseType
- SError
- SforceSession4
- SObject4
- SforceSessionEvents4
- Tab4
- TabSet4
- TabSetCollection4
- UserInfoResult8

For other objects, seeAPI Call Basics.

System Fields

The following fields are read-only fields found on most objects. These fields are automatically updated during API operations. For example, the ID field is automatically generated during a create operation and the LastModifiedDate is automatically updated during any operation on an object.

Field	Field Type	Description
Id	ID	Globally unique string that identifies a record. For information on IDs, see ID Field Type. Because this field exists in every object, it is not listed in the field table for each object. Id fields have Defaulted on create and Filter access.
IsDeleted	boolean	Indicates whether the record has been moved to the Recycle Bin (true) or not (false). Because this field does not appear in all objects, it is listed in the field table for each object.

Field	Field Type	Description
		Audit Fields
CreatedById	reference	ID of the User who created this object. CreatedById fields have Defaulted on create and Filter access.
CreatedDate	dateTime	Date and time when this object was created. CreatedDate fields have Defaulted on create and Filter access.
LastModifiedById	reference	ID of the User who last updated this object. LastModifiedById fields have Defaulted on create and Filter access.
LastModifiedDate	dateTime	Date and time when this object was last modified by a user. LastModifiedDate fields have Defaulted on create and Filter access.
SystemModstamp	dateTime	Date and time when this record was last modified by a user or by a workflow process (such as a trigger). SystemModstamp fields have Defaulted on create and Filter access.

If you import data into Salesforce and need to set the value for an audit field, contact salesforce.com. Once salesforce.com enables this capability for your organization, you can set audit field values for the following objects: Account, CampaignMember, Case, CaseComment, Contact, Idea, IdeaComment, Lead, Opportunity, and Vote. The only audit field you cannot set a value for is systemModstamp.

For information on setting audit fields for custom objects, see Audit Fields for Custom Objects.

These standard objects do not have the audit fields:

Object	No Created byId	No Created Date	No LastMod ifiedById	No LastMod ifiedDate	No System Modstamp	Only System Modstamp (no other audit fields)
AccountShare	Х	Х			Х	
CaseComment			Х	Х		
CaseHistory			Х	Х	Х	
CaseSolution			Х	Х		
CaseShare	Х	Х			Х	
EmailStatus					Х	
GroupMember						Х
LeadShare	Х	Х			Х	
Name					Х	Х
OpportunityHistory			Х	Х		
OpportunityShare	Х	Х			Х	
ProcessInstanceHistory			Х	Х		
ProcessInstanceStep			Х	Х		

Object	No Created byId	No Created Date	No LastMod ifiedById	No LastMod ifiedDate	No System Modstamp	Only System Modstamp (no other audit fields)
ProcessInstanceWorkitem			Х	Х		

Required Fields

Required fields must have a non-null value. This rule affects the Create () and Update () calls:

- When a client application invokes the Create() call, Salesforce automatically populates the data for certain required fields (such as system fields and the object ID fields). Similarly, if a required field has a default value (its DefaultOnCreate property is set to true, as described in Field4, then Salesforce implicitly assigns a value for this field when the object is created, even if a value for this field is not explicitly passed in on the Create() call. For all other required fields, such as ID fields that are analogous to foreign keys in SQL (see Reference Field Type), a client application must explicitly assign a value when the object is created (it cannot be null).
- When a client application invokes the Update () call, a required field cannot be set to null. Many required fields cannot be changed in an Update () call.

Any field not specified as required in the object description is optional, that is, it can be null when updated or created.

For more information about the special handling of required fields for particular objects, see the documentation for such objects later in this topic.

Frequently-Occurring Fields

In addition to System Fields, the following fields are found on many objects:

- OwnerId
- RecordTypeId
- CurrencyIsoCode

OwnerId

Objects have an ownerId field that is an reference to the user who owns that object. Ownership is an important concept that affects the security model and has other implications throughout the system. Any user can query the owner field for any record they can access. However, setting the ownerId field has the following limitations:

- For most users and most objects, this field cannot be set directly upon insert. It is implicitly set to the current user when inserting an object.
- When creating or updating a Case or Lead, a client application (that is logged in with sufficient permissions to transfer a record) can set this field to any valid User in the organization or to any valid queue of the appropriate type in the organization.
- Updating this field via the API changes only the owner of that record. The change of ownership does not cascade to associated records as it does when you transfer record ownership in the Salesforce user interface.
- Updating this field on an account deletes the existing sharing information and reapplies the organization-wide sharing defaults and sharing rules.

In API version 12.0 and later, if your organization has set up sales teams, OwnerId fields behave the same for Account and Opportunity objects as for other objects. That is, if you update the OwnerId field in either object, any AccountShare or OpportunityShare records are kept. In API version 11.0 and earlier, the sharing records are deleted.

RecordTypeId

Record types are used to offer different business processes and subsets of picklist values to different Users based on their Profile settings. (In addition, person accounts use record types to manage a number of additional elements. For more information, see What is a Person Account? in the Salesforce online help.)

Record types are configured in the Salesforce user interface or by creating, editing, or deleting the Record Type object in the API. A client application can also retrieve the list of valid record type IDs (String) for a given object by calling Query() on the Record Type object.

The RecordTypeId field in an object contains the ID of the RecordType that is associated with a standard or custom object. Client applications can set this field in Create() or Update() calls. If specified in a Create() or Update() call, the record type ID (String) must refer to a valid record type for that object. (For more information about ID fields, see ID Field Type.)



Note: The RecordTypeId field is available only if at least one record type is configured for your organization in the Salesforce user interface.

CurrencyIsoCode

For organizations that have multicurrency enabled, the CurrencyIsoCode field contains the string representation of the currency ISO code associated with currency values in the object. Note that the User object also has a DefaultCurrencyIsoCode field, which is the default currency for that user. For example, a user in France could have a DefaultCurrencyIsoCode set to Euros, and that would be their default currency in the application. However, the User object could have currency custom fields stored in a different currency.

API Field Properties

Fields on objects represent the details of each object and are analogous to columns in a database table. Each field on each object has one or more of the following properties:

Property	Description
Autonumber	The API creates an autonumber.
Create	Value for the field can be specified during create using the API.
Defaulted on create	When created, a default value is supplied if no other value is specified.
Delete	Value for the field can be deleted using the API.
Filter	Can be used as filter criteria in a SOQL query FROM or WHERE clause.
Nillable	The field can contain a null value.
Query	The field can be queried with SOQL using the API.
Replicate	The value of the field can be replicated using the API.
Restricted picklist	A picklist that depends on the value of another picklist for the values it displays.
Retrieve	Value of the field can be retrieved using the API.

Property	Description
Search	Can be searched with SOSL using the API.
Update	Can be updated using the API.

Relationships Among Objects

Relationships associate objects with other objects in Salesforce. For example, a relationship can link a custom object to standard object in a related list, such as linking a custom object called Bugs to cases to track product defects associated with customer cases. To view the parent and child relationships among standard objects, see the ERD diagrams in Data Model.



Note: You can use parent-child relationships in SOQL queries. For more information, see Relationship Queries.

You can define different types of relationships by creating custom relationship fields on an object. The differences between relationship types include how they handle data deletion, record ownership, security, and required fields in page layouts:

- Master-Detail (1:n) A parent-child relationship in which the master object controls certain behaviors of the detail object:
 - When a record of the master object is deleted, its related detail records are also deleted.
 - The Owner field on the detail object is not available and is automatically set to the owner of its associated master record. Custom objects on the "detail" side of a master-detail relationship cannot have sharing rules, manual sharing, or queues, as these require the Owner field.
 - The detail record inherits the sharing and security settings of its master record.
 - The master-detail relationship field (which is the field linking the two objects) is required on the page layout of the detail record.

You can define master-detail relationships between two custom objects or between a custom object and a standard object. However, the standard object cannot be on the "detail" side of a relationship with a custom object. In addition, you cannot create a master-detail relationship in which the User or Lead objects are the master.

When you define a master-detail relationship, the custom object on which you are working is the "detail" side. Its data can appear as a custom related list on page layouts for the other object.

• Many-to-many — You can use master-detail relationships to model *many-to-many* relationships between two standard objects, two custom objects, or a custom object and a standard object. A many-to-many relationship allows each record of one object to be linked to multiple records from another object and vice versa. For example, you may have a custom object called "Bug" that relates to the standard case object such that a bug could be related to multiple cases and a case could also be related to multiple bugs. To create a many-to-many relationship, simply create a custom junction object with two master-detail relationship fields, each linking to the objects you want to relate. See the Salesforce online help for details.

Custom objects with two master-detail relationships are supported in API version 11 and later.

• Lookup (1:n) — This type of relationship links two objects together, but has no effect on deletion or security. Unlike master-detail fields, lookup fields are not automatically required. When you define a lookup relationship, data from one object can appear as a custom related list on page layouts for the other object. See the Salesforce online help for details.

To create relationships, use the Salesforce user interface or Salesforce Metadata API. For detailed information, see the Salesforce online help.

In Axis, the element for a relationship will be a query result for parent-to-child relationships or an for child-to-parent relationships.

Relabeling Fields and Tabs and the API

The Salesforce user interface allows you to change the labels on some fields and tabs. Although you cannot relabel fields or tabs using the API, you can retrieve the current values. To do so, issue a Query() call and inspect the label field of the returned QueryResultSet4.

Force.com AppExchange Object Prefixes and the API

If you have an unmanaged package and a managed package version becomes available, the API names of custom fields, custom objects, and Scontrols in the package will change, because a namespace prefix is added to each of these components to make them unique: name_____c becomes prefix____name___c. To move from an unmanaged package to a managed package version of the same application, we advise that you export the data, uninstall the old package, install the new package, review the name changes, and import with the relevant mapping. For details, see the Salesforce online help topic About Managed Packages.

Custom Objects

In the Salesforce user interface, you can extend your organization's Salesforce data by defining custom objects. Custom objects are custom Salesforce database tables that allow you to store information unique to your organization. For custom objects, the Custom flag—a Boolean field in the SObject4—is true.

Client applications with sufficient permissions can invoke API calls on existing custom objects. You can create new custom objects using the metadata WSDL with a client application or using the Force.com IDE. For more information about using the metadata WSDL to create new custom objects, see the Force.com Metadata API Developer's Guide. For more information about the Force.com IDE, see Developer Force.

Use the following topics to understand how the API interacts with custom objects and fields:

- Naming Conventions for Custom Objects
- Relationships Among Custom Objects
- Audit Fields for Custom Objects
- Sharing and Custom Objects
- Managed Packages and API Names

Naming Conventions for Custom Objects

Custom objects have an associated name field that is defined by your Salesforce administrator during setup. Custom objects must have unique names within your organization.

In the API, the names of custom objects are identified by a suffix of two underscores immediately followed by a lowercase "c" character. For example, a custom object labeled "Issue" in the Salesforce user interface is seen as Issue___ c in that organization's WSDL.

Relationships change the naming convention, see Relationships Among Custom Objects for more information.

In order for a custom object record to appear in the Salesforce user interface, its name field must be populated. If you use the API to create a custom object record that does not have a name, then the ID of that record will be used as its name.
Relationships Among Custom Objects

Custom objects relate to other objects and behave just like standard objects, as described in Relationships Among Objects. For example, cascading deletes are supported in custom objects in a Master-Detail relationship.

Custom objects require special treatment so that they can participate in Relationship Queries. For the relationship field name of a custom object, "__r" is appended to the name to create the ID, and "__c" is appended to the name to create the parent object pointer. For example, if the relationship field name is MyRel, the name of the ID becomes MyRelId__r, and the parent object pointer becomes MyRel_c, and the relationship name is MyRel_r. For more information, see Understanding Relationship Names and Custom Objects and Custom Fields.

The following table summarizes whether a standard object can be:

- The master in a master-detail relationship with a custom object. Master-detail relationships involve cascading deletes and sharing rules that are controlled by the parent.
- The lookup in a lookup relationship on a custom object. In other words, whether a custom object can have a lookup to the standard object.
- Extended with custom fields.

Standard Object	Master-Detail	Lookup	Custom Fields
Account	Yes	Yes	Yes
Campaign	Yes	Yes	Yes
Case	Yes	Yes	Yes
Contact	Yes	Yes	Yes
Contract	Yes	Yes	Yes
Event	No	No	Yes
Lead	No	No	Yes
Opportunity	Yes	Yes	Yes
Product2	No	Yes	Yes
Solution	Yes	Yes	Yes
Task	No	No	Yes
User	No	Yes	Yes

Custom objects can also have many-to-many relationships with other custom objects or standard objects. A many-to-many relationship allows each record of one object to be linked to multiple records from another object and vice versa. For more information, see Relationships Among Objects.

Audit Fields for Custom Objects

Custom objects can have the same audit fields as standard objects. Your organization must be API enabled and have the ability to create audit fields enabled, and you must have the "Modify All Data" permission.

When you create a custom object, the four audit fields CreatedById, CreatedDate, LastModifiedById and LastModifiedDate are automatically created and populated for the object. These fields are read only.

Note these restrictions:

- CreatedDate cannot be greater than the LastModifiedDate.
- You cannot set any date field to be greater than the current time.

For more information about audit fields, see System Fields.

Sharing and Custom Objects

A sharing rule object is created for each custom object that does not have a master-detail relationship to another object. They are similar to standard object sharing rules, for example AccountOwnerSharingRule. If the user creating the custom object has the "Manage Users" permission, a sharing rule object is automatically created for it.

Apex sharing reasons can be retrieved by executing a describeSObject() on the custom object's sharing object, and examining the information in the rowCause field. The name of a sharing object for each custom object is of the form: MyObjectName_Share, similar to AccountShare and other standard object sharing objects.

Tags and Custom Objects

When a custom object is created, a Tag object related to it is also created. These object names are of the form: *MyObjectName*__Tag, similar to AccountTag and other standard object tag objects.

Required Fields in Custom Objects

Managed Packages and API Names

If you have an unmanaged package and a managed package version becomes available, the API names of custom fields, custom objects, and Scontrols in the package will change, because a namespace prefix is added to each of these components to make them unique: name_____c becomes prefix_____name____c. To move from an unmanaged package to a managed package version of the same application, we advise that you export the data, uninstall the old package, install the new package, review the name changes, and import with the relevant mapping. For details, see the Salesforce online help topic About Managed Packages.

Chapter 3

API Call Basics

In this chapter ...

- Characteristics of API Calls
- Factors that Affect Data Access
- Synchronous and Asynchronous Calls
- Single-Object and Batch Operations

API calls represent specific operations that your client applications can invoke at runtime to perform tasks, for example:

- Query data in your organization.
- Add, update, and delete data.
- Obtain metadata about your data.
- Run utilities to perform administration tasks.

Using your COM development environment, you can construct Web service client applications that use standard Web service protocols to programmatically:

- Log in to the login server (Login () call) and receive authentication information to be used for subsequent calls
- Query your organization's information (Query (), and Retrieve () calls)
- Perform text searches across your organization's information (Search () call)
- Create, update, and delete data (Create (), and Update (), and Undelete () calls)
- Perform administrative tasks, such as retrieving user information (GetUserInfo() call), changing passwords (SetPassword() and ResetPassword() calls), and getting the system time (GetServerTimestamp() call)
- Replicate data locally (GetDeleted() and GetUpdated() calls)
- Obtain and navigate metadata about your organization's data (DescribeGlobal(), DescribeSObject(), DescribeSObjects(), and DescribeTabs() calls)
- Refresh local objects with the latest data from the database (Refresh () call)
- Set SOAP header options (SetSOAPHeader () call)

See Core Calls, Describe Calls, and Utility Calls for complete details about each call.

Characteristics of API Calls

All API calls are:

- Service Requests and Responses—Your client application prepares and submits a service request to the Force.com Web Service via the API, the Force.com Web Service processes the request and returns a response, and the client application handles the response.
- Synchronous or Asynchronous—Client applications can specify synchronous or asynchronous execution for most API calls.
- Single- or Batch-Object Processing—Client applications can process one or multiple objects in a single API call.
- **Committed Automatically**—Every operation that writes to a Salesforce object is committed automatically. This is analogous to the AUTOCOMMMIT setting in SQL. For Create(), andUpdate() calls that attempt to write to multiple records in an object, the write operation for each record is treated as a separate transaction. For example, if a client application attempts to create two new accounts, they're created using mutually exclusive insert operations that succeed or fail individually, not as a group.

Factors that Affect Data Access

When using the API, the following factors affect access to your organization's data:

- Your organization must be enabled for access, and the user attempting to access the must have the profile permission "API Enabled" selected. It is selected by default.
- Some objects may not appear in the WSDL because you must first contact salesforce.com and request access, for exampleTerritory or Campaign. These objects are noted in the "Usage" section for each object.
- Whether your configured permissions allow access to the data. Your client application logs in as a user to the Force.com Web Service. The profile associated with the logged-in user grants or denies access to specific objects and fields in your organization.

For certain objects, the user profile is configured with one of the following permissions:

- Read—Users can only view objects of this type.
- Create—Users can read and create objects of this type.
- Edit—Users can read and update objects of this type.
- Delete—Users can read, edit, and delete objects of this type.

These permissions apply to many objects, including: Account, Asset, Campaign, Case, Contact, Contract, Document, Lead, Opportunity, Pricebook2, Product2, and Solution. Other objects (such as AccountTeamMember) follow sharing on the associated permission-assigned object (such as the Account record). Similarly, a Partner depends on the permissions in the associated Account.

There are also permissions that override sharing:

- View All-users can view all records associated with this object, regardless of sharing settings
- Modify All—users can read, edit, delete, transfer, and approve all records associated with this object, regardless of sharing settings
- Modify All Data—users can read, edit, delete, transfer, and approve all records regardless of sharing settings. This permission is not an object-level permission, unlike "View All" and "Modify All."

When an application logs into the API, all transactions are run as the user who logs in. Therefore, to protect the security of your data, give that user (the logged-in user) only the permissions needed to successfully execute all the calls made by the application. For large integration applications, "Modify All Data" may be required.

User permissions do not affect field-level security. If field-level security specifies that a field is hidden, users with "Read" on that object can view only those fields that are not hidden on the record. In addition, users with "Read" on an object can view only those records that sharing settings allow. The one exception is the "Edit Read Only Fields" permission, which gives users the ability to edit fields marked as read only via field-level security.

The API respects object-level and field-level security configured in the Salesforce user interface. You can access objects and fields only if the security settings in the logged-in user's profile permit such access. For example, fields that are not visible to a given user are not returned in a Query() or DescribeSObjects() call.

- Whether the sharing model of the logged-in user allows access to the data. For most API calls, data that is outside of the logged-in user's sharing model is not returned. Users are granted the most permissive access that is available to them, either through organization-wide defaults or manual record sharing.
- Whether your organization uses territory management to control user access to accounts, cases, and opportunities. Territory management is available in Enterprise, Unlimited, and Developer Edition organizations. For more information, see the Salesforce online help.
- Whether a given object is configured to be accessible via certain API calls. For example, to create an object via the Create() call, its object must be configured as createable (the value for the object's createable attribute is set to true) in the SObject4. To determine what operations are allowed on a given object, your client application can invoke the DescribeSObjects() call on the object and inspect the following properties in the DescribeSObjectResult: creatable (Create() call), updateable (Update() call), queryable (Query() call), retrievable (Retrieve() call), searchable (Search() call), and replicatable (GetUpdate() and GetDeleted() calls).
- Whether a particular change would compromise the referential integrity of your organization's Salesforce data. For example:
 - ID values in reference fields (see Reference Field Type) are validated in Create () and Update () calls. For information on IDs, see ID Field Type.
 - If a client application deletes an object instance, then its children are automatically deleted as part of the call if the CascadeDelete property on ChildRelationship4 for that child has a value of true. For example, if a client application deletes an Opportunity, then any associated OpportunityLineItem records are also deleted. Use SObject4 to view the ChildRelationship4 value if you want to be sure what will be deleted.

There are certain exceptions which prevent the execution of a CascadeDelete. For example, you cannot delete an account if it has associated cases, if it has related opportunities that are owned by other users, or if associated contacts are enabled for the Self-Service portal. In addition, if you attempt to delete an account that has closed/won opportunities owned by you or has active contracts, then the delete request for that record will fail.

- Whether a given field in an object can be updated or not. For example, read-only fields cannot be changed in Create() or Update() calls.
- Whether a given feature is used by your organization. For example, record type IDs are available only if at least one record type is configured for your organization in the Salesforce user interface.
- Rules for custom objects—such as fields that are configured in the Salesforce user interface to be unique—are not yet enforced via the API, however rules that specify that a field in a custom object is required are enforced. For more information, see Required Fields in Custom Objects.
- Ownership changes to one object instance do not automatically cascade to other object instances. For example, if ownership changes for a given Account, ownership does not then automatically change for any Contract associated with that Account—each ownership change must be made separately and explicitly by the client application.
- Certain features that are configurable in the Salesforce user interface are not accessible or implicitly enforced via the API. For example:
 - Page layouts can specify whether a given field is required, but the API does not enforce such layout-specific field restrictions or validations in Create() and Update() calls. It is up to the client application to enforce any such constraints, if applicable.
 - Record types can control which picklist values can be chosen in a given record and which page layouts users with different profiles can see. However, such rules that are configured and enforced in the Salesforce user interface are not enforced in the API. For example, the API will not validate whether the value in a picklist field is allowed per any record type restrictions associated with the profile of the logged-in user. Similarly, the API will not prevent a client

application from adding data to a particular field simply because that field does not appear in a layout associated with the profile of the logged-in user.

If any such constraints are required, it is up to business logic in the client application to enforce them explicitly.

Synchronous and Asynchronous Calls

The API supports synchronous and asynchronous processing. All API calls support synchronous invocation, in which the client application waits for a response from the API before continuing.

Asynchronous calls allow client applications to continue doing other work while awaiting a response from the API. The following API calls (batch operations only) in the Office Toolkit support asynchronous invocation:Create(), Update(), Query(), Retrieve(), Search(), and Refresh(); GetDeleted(), and GetUpdated(). These calls require a Boolean async parameter that specifies whether the call is to be processed asynchronously(true) or synchronously(false). For asynchronous invocations, the appropriate callback function (see Events) on the SforceSession4 object is triggered when the Force.com Web Service returns the result to the client application. Client applications must provide the handler code in the callback function.

For code examples of each type of operation, see:

- Query—Synchronous Example
- Query—Asynchronous Example
- Retrieve—Synchronous Example
- Retrieve—Asynchronous Example
- Update—Synchronous Example
- Update—Asynchronous Example

Single-Object and Batch Operations

The API provides different ways to work with objects:

- Single-object operations that work on single object instances on page 40
- Batch operations that work on batches of objects (one or more objects) on page 41

Single-Object Operations

When working with a single object instance (such as creating one new Account), client applications can use the following API operations:

- CreateObject() to create a new, empty SObject4 instance.
- Functions on the SObject4 object that Create, Update, or Delete this object instance in the organization's Salesforce data, as well as Refresh to refresh the local SObject4 instance with the latest Salesforce data, and Clone to create an exact copy of the SObject4 instance.

All single-object operations are process synchronously. Only batch operations can be called asynchronously.

The following code example shows a typical sequence of tasks to add a new Account using single-object operations, including:

- 1. Calling CreateObject () to create a new local instance of an Account.
- 2. Specifying an account name (and otherwise populating fields in the SObject4 with data).
- 3. Calling the Create function on that account to commit the changes in the organization's Salesforce data.

The code example demonstrates other optional tasks, such as calling the Refresh function on the SObject4 instance to refresh the local object instance with the latest information from the organization's Salesforce data.

```
'Create a blank object
Public Function CreateAccount() As SObject4
On Error GoTo handleError
'all entities are of type SObject4
  Dim account As SObject4
'create a blank object
 Set account = g sfApi.CreateObject("account")
    account("name") = "Sample"
'commit the object
 account.Create
'refresh the object to get server set values
    account.Refresh
'set as the function return value
    Set CreateAccount = account
handleError:
MsgBox g sfApi.ErrorMessage
End Function
```

Batch Operations

Client applications use batch operations to work with one or more object instances, such as creating multiple Account records. Batch operations (Create() and Update() calls) accept an array of one or more SObject4 instances and provide the option of synchronous or asynchronous processing (see Synchronous and Asynchronous Calls on page 40). When processing multiple objects, it is more efficient and faster to process them in a single batch operation than in multiple single-object operations.

The following code example shows a typical call instance for creating a new Account using batch operations:

- **1.** Dimensioning an SObject4 array.
- 2. Specifying an account name for the first element in the array, and otherwise populating fields in the SObject4 element with data.
- 3. Calling Create() (synchronously) to create the new Account in the organization's Salesforce data.
- 4. Checking for the success or failure of the Create () call, and handling any errors.

```
Public Function CreateAccount() As SObject4
On Error GoTo handleError
'create using batch call
Dim so(0) As SObject4
```

```
Set so(0) = g_sfApi.CreateObject("account")
    so(0) ("name") = "Test Array account"
'call batch method
   g_sfApi.Create so, False
'call batch refresh
g sfApi.Refresh so, False
   Exit Function
handleError:
 MsgBox g_sfApi.ErrorMessage
If g_sfApi.Error <> NO_SF_ERROR Then
'This does not mean all of the SObject4s in the array failed,
'only that one or more have failed.
'Check each object to determine if there is an error.
 Debug.Print g_sfApi.ErrorMessage
  'look at the array
       For i = 0 To UBound(so)
       If so(i).Error <> NO_SF_ERROR Then
       Debug.Print so(i).ErrorMessage
 'fix the error and call update or create again on the object?
      so(i)("name") = "callback mod"
      so(i).Update
       Next i
End If
End Function
```

Chapter 4

Error Handling

In this chapter ...

- Error Handling for Session Expiration
- More About Error Handling
- Sample Error Handling—VBA

The API calls return error data that your client application can use to identify and resolve runtime errors. If an error occurs during the invocation of most API calls, then the API returns an array of one or more SErrors and error message text.

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.

More About Error Handling

For more information about errors, see SError.

Sample Error Handling—VBA

The following sample code shows a basic error handler function that returns true if no error occurred (NO_SF_ERROR) during the execution of an API call. If an error occurred, the function displays the error message in a message box and returns false.

```
Function CheckApiError() As Boolean
CheckApiError = g_sfApi.Error = NO_SF_ERROR
If g_sfApi.Error <> NO_SF_ERROR Then
MsgBox g_sfApi.ErrorMessage
End If
End Function
```

Chapter 5

Security and the API

In this chapter ...

- User Authentication
- User Profile Configuration
- Security Token
- Sharing
- Implicit Restrictions for Objects and Fields
- API Access in Force.com AppExchange Packages
- Outbound Port Restrictions

Client applications that access your organization's Salesforce data are subject to the same security protections that are used in the Salesforce user interface. Additional protection is available for organizations that install Force.com AppExchange managed packages if those packages contain components that access Salesforce via the API.

User Authentication

Client applications must log in using valid credentials for an organization. The server authenticates these credentials and, if valid, provides the client application with: a SessionId on page 343. A client application can check the IsLoggedIn property in the SforceSession4 object, returned after a successful Login () call, to verify that the session is still valid.

Salesforce.com supports only the Secure Sockets Layer (SSL) protocol SSLv3 and the Transport Layer Security (TLS) protocol. Ciphers must have a key length of at least 128 bits.

User Profile Configuration

An organization's Salesforce administrator controls the availability of various features and views by configuring profiles and assigning users to them. To access the API (to issue calls and receive the call results), a user must be granted the "API Enabled" profile permission. Client applications can query or update only those objects and fields to which they have access via the profile of the logged-in user.

To create, edit, or delete a profile, go to Setup > Manage Users > Profiles in the Salesforce user interface.

The EntityNames property in the SforceSession4 object, returned after a successful Login () call, provides a list of available objects for an organization.

Security Token

When users log in to Salesforce, either via the user interface, the API, or a desktop client such as Connect for Outlook, Connect Offline, Connect for Office, Connect for Lotus Notes, or the Data Loader, Salesforce confirms that the login is authorized as follows:

- 1. Salesforce checks whether the user's profile has login hour restrictions. If login hour restrictions are specified for the user's profile, any login outside the specified hours is denied.
- 2. Salesforce then checks whether the user's profile has IP address restrictions. If IP address restrictions are defined for the user's profile, any login from an undesignated IP address is denied, and any login from a specified IP address is allowed.
- 3. If profile-based IP address restrictions are not set, then Salesforce checks whether the user is logging in from an IP address they have not used to access Salesforce before:
 - If the user's login is from a browser that includes a Salesforce cookie, the login is allowed. The browser will have the Salesforce cookie if the user has previously used that browser to log in to Salesforce, and has not cleared the browser cookies.
 - If the user's login is from an IP address in your organization's trusted IP address list, the login is allowed.
 - If the user's login is from neither a trusted IP address nor a browser with a Salesforce cookie, the login is blocked.

Whenever a login is blocked or returns an API login fault, Salesforce must verify the user's identity:

• For access via the user interface, the user is prompted to click a **Send Activation Link** button to send an activation email to the address specified on the user's Salesforce record. The email instructs the user to copy and paste an activation link into their browser to activate their computer for logging in to Salesforce. The activation link included in the email is valid for up to 24 hours from the time the user clicked the **Send Activation Link** button. After 24 hours, the activation link expires, and users must repeat the activation process to log in.

• For access via the API or a client, the user must add their security token to the end of their password in order to log in. A security token is an automatically-generated key from Salesforce. For example, if a user's password is mypassword, and their security token is XXXXXXXXX, then the user must enter mypasswordXXXXXXXXX to log in.

Users can obtain their security token by changing their password or resetting their security token via the Salesforce user interface. When a user changes their password or resets their security token, Salesforce sends a new security token to the email address on the user's Salesforce record. The security token is valid until a user resets their security token, changes their password, or has their password reset.



Tip: It is recommended that you obtain your security token via the Salesforce user interface from a trusted network prior to attempting to access Salesforce from a new IP address.



Note: For more information about tokens, see "Resetting Your Security Token" in the Salesforce online help.

When a user's password is changed, the user's security token is automatically reset. The user will experience a blocked login until he or she adds the automatically-generated security token to the end of his or her password or enters the new password after the administrator adds their IP address to the organization's list of trusted IP addresses.

If Single Sign-On (SSO) is enabled for your organization, users who access the API or a desktop client cannot log in to Salesforce unless their IP address is included on your organization's list of trusted IP addresses or on their profile, if their profile has IP address restrictions set. Futhermore, the delegated authentication authority usually handles login lockout policies for users with the "Uses Single Sign-On" permission. However, if the security token is enabled for your organization, then your organization's login lockout settings determine the number of times a user can attempt to log in with an invalid security token before being locked out of Salesforce. For more information, see "Setting Login Restrictions" and "Setting Password Policies" in the online help.

Sharing

In the Salesforce user interface, sharing refers to the act of granting read or write access to a user or group so that they can view or edit a record owned by other users, if the default organization access levels do not otherwise permit such access. All API calls respect the sharing model.

The following table describes the types of access levels.

API Value	Salesforce User Interface Label	API Picklist Label	Description
None	Private	Private	Only the record owner and Users above that role in the hierarchy can view and edit the record.
Read	Read Only	Read Only	All Users and Groups can view the record but not edit it. Only the owner and users above that role in the hierarchy can edit the record.
Edit	Read/Write	Read/Write	All Users and Groups can view and edit the record.
ReadEditTransfer	Read/Write/Transfer	Read/Write/Transfer	All Users and Groups can view, edit, delete, and transfer the record. (Only available for cases and leads as an organization-wide default setting.)

API Value	Salesforce User Interface Label	API Picklist Label	Description
All	Full Access	Owner	All Users and Groups can view, edit, transfer, delete, and share the record. (Only available for campaigns as an organization-wide default setting.)
ControlledByParent	Controlled by Parent	Controlled By Parent	(Contacts only.) All Users and Groups can perform an action (such as view, edit, or delete) on the contact based on whether he or she can perform that same action on the record associated with it.

Not all access levels are available for every object. See the Fields table for each object to learn which access levels are available, as well as other sharing details specific to that object.

For more information about sharing in general, see the Salesforce online help.



Note: In the API, you can create and update objects such as AccountShare and OpportunityShare that define sharing entries for records.

Implicit Restrictions for Objects and Fields

Certain objects can be created or deleted only in the Salesforce user interface. Other objects are read-only—client applications cannot Create() or Update() such objects. Similarly, certain fields within some objects can be specified on Create() but not on Update(). Other fields are read-only—client applications cannot specify field values in Create() or Update() calls. For more information, see the respective object descriptions in Standard and Custom Object Basics.

API Access in Force.com AppExchange Packages

The API allows access to objects and calls based on the permissions of the user who logs into the API. To prevent security issues from arising when installed packages have components that access data via the API, Salesforce provides additional security:

- When a developer creates an AppExchange package with components that access the API, the developer can restrict the API access for those components.
- When an administrator installs an AppExchange package, the administrator can accept or reject the access. Rejecting the access cancels the installation.
- After an administrator installs a package, the administrator can restrict the API access of components in the package that access the API.

Editing API access for a package is done in the Salesforce user interface. For more information, see "Managing API and Dynamic Apex Access in Packages" in the Salesforce online help.

API access for a package affects the API requests originating from components within the package; it determines the objects that the API requests can access. If the API access for a package is not defined, then the objects that the API requests have access to are determined by the user's profile permissions.

The API access for a package never allows users to do more than the permissions granted on the user's profile. API access in a package only reduces what the user's profile allows.

Choosing Restricted for the API Access setting in a package affects the following:

- API access in a package overrides the following permissions granted in a user's profile:
 - Author Apex
 - Customize Application
 - Edit HTML Templates
 - Edit Read Only Fields
 - Manage Billing
 - Manage Call Centers
 - Manage Categories
 - Manage Custom Report Types
 - Manage Dashboards
 - Manage Letterheads
 - Manage Package Licenses
 - Manage Public Documents
 - Manage Public List Views
 - Manage Public Reports
 - Manage Public Templates
 - Manage Users
 - Transfer Record
 - Use Team Reassignment Wizards
 - View Setup and Configuration
 - Weekly Export Data
- If Read, Create, Edit, and Delete access are not selected in the API access setting for objects, users do not have access to those objects from the package components, even if the user has the "Modify All Data" and "View All Data" permissions.
- Salesforce denies access to Web service and executeanonymous requests from an AppExchange package that has Restricted access. For more information about executeanonymous and accessing Web services with Apex, see the *Apex Language Reference*.

The following considerations also apply to API access in packages:

- · Workflow rules and Apex triggers fire regardless of API access in a package.
- If a component is in more than one package in an organization, API access is unrestricted for that component in all packages in the organization regardless of the access setting.
- If Salesforce introduces a new standard object after you select restricted access for a package, access to the new standard object is not granted by default. You must modify the restricted access setting to include the new standard object.
- When you upgrade a package, changes to the API access are ignored even if the developer specified them. This ensures that the administrator installing the upgrade has full control. Installers should carefully examine the changes in package access in each upgrade during installation and note all acceptable changes. Then, because those changes are ignored, the administrator should manually apply any acceptable changes after installing an upgrade.
- S-controls are served by Salesforce and rendered inline in Salesforce. Because of this tight integration, there are several means by which an s-control in an installed package could escalate its privileges to the user's full privileges. In order to protect the security of organizations that install packages, s-controls have the following limitations:
 - For packages you are developing (that is, not installed from AppExchange), you can only add s-controls to packages with the default Unrestricted API access. Once a package has an s-control, you cannot enableRestricted API access.

- For packages you have installed, you can enable access restrictions even if the package contains s-controls. However, access restrictions provide only limited protection for s-controls. Salesforce recommends that you understand the JavaScript in an s-control before relying on access restriction for s-control security.
- If an installed package has Restricted API access, upgrades will be successful only if the upgraded version does not contain any s-controls. If s-controls are present in the upgraded version, you must change the currently installed package to Unrestricted API access.

To manage API access to packages, see "Managing API and Dynamic Apex Access in Packages" in the Salesforce online help.

Note: XML-RPC requests that originate from restricted packages will be denied access.

Outbound Port Restrictions

For security reasons, Salesforce restricts the outbound ports you may specify to one of the following:

- 80: This port only accepts HTTP connections.
- 443: This port only accepts HTTPS connections.
- 7000-10000 (inclusive): These ports accept HTTP or HTTPS connections.

The port restriction applies to any feature where a port is specified, for example outbound messages, AJAX proxy, or single-sign on.

REFERENCE

Chapter 6

Data Model

The entity relationship diagrams (ERDs) for standard Salesforce objects in this section illustrate important relationships between objects. The available ERDs are:

- · Sales Objects-includes accounts, contacts, opportunities, leads, campaigns, and other related objects
- · Task and Event Objects-includes tasks and events and their related objects
- · Support Objects-includes cases and solutions and their related objects
- · Document, Note, and Attachment Objects-includes documents, notes, and attachments and their related objects
- User and Profile Objects-includes users, profiles, and roles
- · Record Type Objects-includes record types and business processes and their related objects
- · Product and Schedule Objects-includes opportunities, products, and schedules
- · Sharing and Team Selling Objects-includes account teams, sales teams, and sharing objects
- · Territory Management-includes territories and related objects

Each entity relationship diagram includes links to the topics that describe the fields in objects related to the diagram. The data model for your custom objects depends on what you create.

Sales Objects



Task and Event Objects



Support Objects



Document, Note, and Attachment Objects



User and Profile Objects



Note: All OwnerId, CreatedById, and LastModifiedById fields on other objects relate to the User object.

Record Type Objects



Product and Schedule Objects



Create a separate PricebookEntry for each currency and price combination.

Sharing and Team Selling Objects



Territory Management



Process Objects



Chapter 7

Standard Objects

This section provides a list of standard objects and their standard fields. Each list does not include the System Fields, and may not include all fields for your particular organization, depending on which features are enabled.

A client application can inspect an SObject4 of this type, iterating through the array of Field4 objects in the Fields property, to verify the complete list of fields.

The API provides access to the following standard objects:

Object	Description
Account	An individual account, which is an organization involved with your business (such as customers, competitors, and partners).
AccountContactRole	The role that a given Contact plays on an Account.
AccountHistory	Represents the history of changes to the values in the fields of an account. This object is available in versions 11.0 and later.
AccountOwnerSharingRule	A rule that grants access to an account to users other than the owner.
AccountPartner	A relationship between two Account objects, such as partnerships or subsidiaries.
AccountShare	A sharing entry on an Account.
AccountTag	Associates a word or short phrase with an Account.
AccountTeamMember	A User who is a member of an Account team.
AccountTerritoryAssignmentRule	A rule that assigns accounts to territories.
AccountTerritoryAssignmentRuleItem	A field-specific criteria row for an AccountTerritoryAssignmentRule.
AccountTerritorySharingRule	Rules for sharing an account within a territory.
ActivityHistory	Information about tasks and events related to an object.
AdditionalNumber	An additional phone number for a CallCenter.
ApexClass	Represents an Apex class. For information, see the <i>Apex Language Reference</i> .
ApexComponent	A Visualforce custom component that can be used in a Visualforce page alongside standard components

Object	Description
	<pre>such as <apex:relatedlist> and <apex:datatable>.</apex:datatable></apex:relatedlist></pre>
ApexPage	A Visualforce page, containing Visualforce markup, HTML, Javascript, and other Web-enabled code.
ApexTrigger	Represents an Apex trigger.
Approval	An approval request for a Contract.
Asset	An asset (such as product previously sold and installed) owned by an Account or Contact.
AssetTag	Associates a word or short phrase with an Asset.
AssignmentRule	An assignment rule associated with a Case or Lead.
Attachment	A file that a User has uploaded and attached to a parent object.
Bookmark	A link between two opportunities.
BrandTemplate	Letterhead for email templates.
BusinessHours	Specifies the business hours of your support organization. Escalation rules are run only during these hours.
BusinessProcess	A business process.
CallCenter	A single computer telephony integration (CTI) system instance in an organization.
Campaign	A marketing campaign, such as a direct mail promotion, webinar, or trade show.
CampaignMember	The association between a Campaign and either a Lead or Contact.
CampaignMemberStatus	A status value associated with a Campaign.
CampaignOwnerSharingRule	Represents the rules for sharing a Campaign with Users other than the owner.
CampaignShare	Represents a list of access levels to a Campaign along with an explanation of the access level. For example, if you have access to a record because you own it, the Access Level value is Full and Reason for Access value is Owner.
CampaignTag	Associates a word or short phrase with a Campaign.
Case	A customer issue such as a customer's feedback, problem, or question.
CaseComment	A comment that provides additional information about the associated Case.
CaseContactRole	The role that a given Contact plays on a Case.

Object	Description
CaseHistory	Historical information about changes that have been made to the associated Case.
CaseOwnerSharingRule	A rule that grants access to a case to users other than the owner.
CaseShare	A sharing entry on a Case.
CaseSolution	The association between a particular Case and a particular Solution.
CaseStatus	The status of a Case, such as New, On hold, In Process, and so on.
CaseTag	Associates a word or short phrase with a Case.
CaseTeamMember	Represents a case team member, who works with a team of other users to help resolve a case.
CaseTeamRole	Represents a case team role. Every case team member has a role on a case, such as "Customer Contact" or "Case Manager."
CaseTeamTemplate	Represents a predefined case team, which is a group of users that helps resolve a case.
CaseTeamTemplateMember	Represents a member on a predefined case team, which is a group of users that helps resolve cases.
CaseTeamTemplateRecord	The CaseTeamTemplateRecord object is a linking object between the Case and CaseTeamTemplate objects. To assign a predefined case team to a case (customer inquiry), create a CaseTeamTemplateRecord object and point the ParentId to the case and the TeamTemplateId to the predefined case team.
CategoryData	A logical grouping of Solution records.
CategoryNode	A tree of Solution categories.
CategoryNodeLocalization	The translated value of the label for a catgeory.
Community	Represents a community within Salesforce CRM Ideas.
Contact	A contact, which is an individual associated with an Account.
ContactHistory	Represents the history of changes to the values in the fields of a contact. This object is available in versions 11.0 and later.
ContactOwnerSharingRule	Represents the rules for sharing a contact with Users other than the owner.
ContactShare	Represents a list of access levels to a Contact along with an explanation of the access level. For example, if you have access to a record because you own it,

Object	Description
	the Access Level value is Full and Reason for Access value is Owner.
ContactTag	Associates a word or short phrase with a Contact.
Contract	A contract (a business agreement) associated with an Account.
ContractContactRole	The role that a given Contact plays on a Contract.
ContractHistory	Information about changes to a contract.
ContractStatus	The status of a Contract, such as Draft, In Approval, Activated, Terminated, or Expired.
ContractTag	Associates a word or short phrase with a Contract.
CurrencyType	The currencies used by an organization for which the multicurrency feature is enabled.
DatedConversionRate	The effective dated exchange rates used by an organization for which advanced currency management is enabled.
Division	A logical segment of your organization's data.
Document	A file that a user has uploaded. Unlike Attachment objects, Documents are not attached to a parent object.
DocumentAttachmentMap	Maps the relationship between an EmailTemplate and its attachment, which is stored as a Document.
DocumentTag	Associates a word or short phrase with a Document.
EmailMessage	An email message related to Email-to-Case.
EmailServicesAddress	An email service address.
EmailServicesFunction	An email service.
EmailStatus	The status of an email sent via Salesforce.
EmailTemplate	A template for sending email via Salesforce.
EntityHistory	Removed as of 8.0. Use the history object corresponding to the object you are working with.
Event	A calendar appointment event.
FiscalYearSettings	Fiscal year settings.
Folder	A repository for a Document, MailMergeTemplate, email template, or report. Only one type of item can be contained in a particular Folder.
ForecastShare	Represents a sharing entry of a forecast at a given role and territory.
Group	A set of User records.
GroupMember	A User or Group that is a member of a public group.

Object	Description
Holiday	Represents a period of time during which your customer support team is unavailable. Business hours and escalation rules associated with business hours are suspended during any holidays with which they are affiliated.
Idea	Represents an idea on which users are allowed to comment and vote, for example, a suggestion for an enhancement to an existing product or process.
IdeaComment	Represents a comment that a user has submitted in response to an idea.
Lead	A lead, which is a prospect or potential Opportunity.
LeadHistory	Represents the history of changes to the values in the fields of a lead.
LeadOwnerSharingRule	Rules that assign an owner to a lead.
LeadShare	A sharing entry on a Lead.
LeadStatus	The status of a Lead, such as Open, Qualified, or Converted.
LeadTag	Associates a word or short phrase with a Lead.
LineitemOverride	Customizable forecast data for an opportunity line item.
MailMergeTemplate	A mail merge template (a Microsoft Word document) used for performing mail merges for your organization.
Name	Non-queryable object that provides information about foreign key traversals when the foreign key has more than one parent.
Note	A note, which is text associated with an Attachment, Contact, Contract, Opportunity, or custom object.
NoteTag	Associates a word or short phrase with a Note.
NoteAndAttachment	Information about the notes and attachments for an object.
OpenActivity	Information about the open tasks and events for an object.
Opportunity	An opportunity, which is a sale or pending deal.
OpportunityCompetitor	A competitor on an Opportunity.
OpportunityContactRole	The role that a Contact plays on an Opportunity.
OpportunityFieldHistory	The history of changes to the field values of an Opportunity. This object is available in versions 13.0 and later
OpportunityHistory	The stage history of an Opportunity.

Object	Description
OpportunityLineItem	An opportunity line item, which is a member of the list of Product2s associated with an Opportunity, along with other information about those products on that opportunity.
OpportunityLineItemSchedule	Information about the quantity, revenue distribution, and delivery dates for a particular OpportunityLineItem.
OpportunityOverride	Customizable forecast data for an opportunity.
OpportunityOwnerSharingRule	A rule that grants access to an opportunity to users other than the owner.
OpportunityPartner	A partner relationship between an Account and an Opportunity.
OpportunityShare	A sharing entry on an Opportunity.
OpportunityStage	The stage of an Opportunity in the sales pipeline, such as New Lead, Negotiating, Pending, Closed, and so on.
OpportunityTag	Associates a word or short phrase with an Opportunity.
OpportunityTeamMember	An individual User on the sales team of a particular Opportunity.
Organization	A business, company, or other organization.
Partner	The association between two particular accounts or between a particular Opportunity and an Account.
PartnerNetworkConnection	A connection in Salesforce to Salesforce.
PartnerNetworkRecordConnection	A record that is shared with a connection using Salesforce to Salesforce.
PartnerRole	A role for an account Partner, such as a consultant or supplier.
Period	A fiscal period.
Pricebook2	A price book that contains the list of Product2s that your organization sells.
PricebookEntry	A product entry (an association between a Pricebook2 and Product2) in a price book.
ProcessInstance	An instance of a single, end-to-end approval chain.
ProcessInstanceHistory	History of changes to a process instance.
ProcessInstanceStep	One step in an approval workflow process instance.
ProcessInstanceWorkitem	A pending approval request to a specific user.
Product2	A product that your organization sells. A product is member of the list of items in a Pricebook2.

Object	Description
Profile	A profile, which defines a set of user permissions for performing different operations, such as querying, adding, updating, or deleting information.
QuantityForecast	A quantity-based customizable forecast.
QuantityForecastHistory	Historical information about a quantity-based customizable forecast.
QueueSobject	Associates SObject4 with a specified queue.
RecordType	A record type.
RevenueForecast	A revenue-based customizable forecast.
RevenueForecastHistory	Historical information about a revenue-based customizable forecast.
Scontrol	A custom s-control, which is custom content that is hosted by the system but executed by the client application.
ScontrolLocalization	The translated value of the field label for an s-control.
SelfServiceUser	A Contact who has been enabled to use your organization's Self-Service portal, where he or she can obtain online support.
Solution	A detailed description of a customer issue and the resolution of that issue.
SolutionHistory	History of changes to a solution.
SolutionStatus	The status of a Solution, such as Draft, Reviewed, and so on.
SolutionTag	Associates a word or short phrase with a Solution.
StaticResource	A static resource that can be used in Visualforce markup.
TagDefinition	Defines the attributes of child Tag objects.
Task	An activity or to-do item to perform or that has been performed.
TaskPriority	The priority (importance) of a Task, such as High, Normal, or Low.
TaskStatus	The status of a Task, such as Not Started, Completed, or Closed.
TaskTag	Associates a word or short phrase with a Task.
Territory	A territory to which users and accounts are assigned.
User	A user in your organization.
UserAccountTeamMember	A single User on the default account team of another user.

Object	Description
UserLicense	A user license in your organization.
UserPreference	A functional preference for a user in your organization.
UserRole	A role in your organization.
UserTeamMember	A single User on the default sales team of another user.
UserTerritory	A single user who has been assigned to a territory.
Vote	Represents a vote that a user has made on an idea.
WebLink	A Web link to an URL or Scontrol.
WebLinkLocalization	The translated value of the field label for a custom link to a URL or s-control.



Note: The Product and Pricebook objects, previously deprecated, have been removed. Requests containing them will be refused, and responses will not contain them.

Account

Represents an individual account, which is an organization or person involved with your business (such as customers, competitors, and partners).

Corresponds to an SObject4 in which the ObjectType="Account".

Supported Calls

```
Create(),Update(),Query(),Search(),Retrieve(),CreateObject(),GetDeleted(),GetUpdated(),
DescribeSObjects()
```

Special Access Rules

Customer Portal users can access their own accounts and any account shared with them.

Fields

Field Name	Field Type	Field Properties	Description
AccountNumber	string	Create Filter Nillable Update	Account number assigned to this account (not the unique, system-generated ID assigned by Salesforce during creation). Maximum size is 40 characters.

Field Name	Field Type	Field Properties	Description
AnnualRevenue	currency	Create	Estimated annual revenue of the account.
		Filter	
		Nillable	
		Update	
• BillingCity	string	Create	Details for the billing address of this account. Maximum size for city and country is 40 characters, maximum size for postal code and state is 20 characters.
• BillingCountry		Filter	
BillingPostalCodeBillingState		Nillable	
		Update	
BillingStreet	textarea	Filter	Street address for the billing address of this account.
		Nillable	
		Update	
ConnectionReceivedID	reference	Filter	ID of the PartnerNetworkConnection that shared this record with your organization. This field is only available if you have enabled Salesforce to Salesforce.
		Nillable	
ConnectionSentID	reference	Filter	ID of the PartnerNetworkConnection that you shared this
		Nillable	Salesforce to Salesforce. Beginning with API version 15.0, the ConnectionSentID field is no longer supported. The ConnectionSentID field is still be visible, but the value is null. You can use the new PartnerNetworkRecordConnection object to forward records to connections.
Description	textarea	Create	Text description of the account. Limited to 32,000 KB.
		Filter	
		Nillable	
		Retrieve	
		Update	
Fax	phone	Create	Fax number for the account.
		Filter	
		Nillable	
		Update	
Industry	picklist	Create	An industry associated with this account.
		Filter	
		Nillable	
		Update	

Field Name	Field Type	Field Properties	Description
IsCustomerPortal	boolean	Defaulted on create Filter	Indicates whether the account has at least one contact enabled to use the organization's Customer portal (true) or not (false). This field is not available until at least on Customer portal has been defined.
IsDeleted	boolean	Defaulted on create Filter	Indicates whether the object has been moved to the Recycle Bin (true) or not (false). Label is Deleted .
IsPartner	boolean	Defaulted on create Filter	Indicates whether the account has at least one contact enabled to use the organization's PRM Portal (true) or not (false). This field is not available until at least on PRM Portal has been defined.
IsPersonAccount	boolean	Defaulted on create Filter	Read only. Label is Is Person Account . Indicates whether this account has a record type of Person Account (true) or not (false). For more information about person accounts, see Person Account Record Types, and the Salesforce online help.
LastActivityDate	date	Nillable	 Value is one of the following, whichever is the most recent: Due date of the most recent event logged against the record. Due date of the most recently closed task associated with the record.
MasterRecordId	reference	Filter Nillable	If this object was deleted as the result of a merge, this field contains the ID of the record that was kept. If this object was deleted for any other reason, or has not been deleted, the value is null.
Name	string	Create Filter Update	 Required. Label is Account Name. Name of the account. Maximum size is 255 characters. If the account has a record type of Person Account: This value is the concatenation of the FirstName and LastName of the associated person contact. You cannot modify this value.
NumberOfEmployees	int	Create Filter Nillable Update	Label is Employees . Number of employees working at the company represented by this account. Maximum size is eight digits.
OwnerId	reference	Create Defaulted on create Filter Update	The ID of the user who currently owns this account. Default value is the user logged in to the API to perform the create. If you have set up account teams in your organization, updating this field has different consequences depending on your version of the API:.

Field Name	Field Type	Field Properties	Description
			 For API version 12.0 and later, sharing records are kept, as they are for all objects. For API version before 12.0, sharing records are deleted.
Ownership	picklist	Create	Ownership type for the account, for example Private, Public, or Subsidiary.
		Filter	
		Nillable	
		Update	
ParentId	reference	Create	ID of the parent object, if any.
		Filter	
		Nillable	
		Update	
Phone	phone	Create	Phone number for this account. Maximum size is 40 characters.
		Filter	
		Nillable	
		Update	
Rating	picklist	Create	The account's prospect rating, for example Hot, Warm, or Cold.
		Filter	
		Nillable	
		Update	
RecordTypeId	reference	Create	ID of the record type assigned to this object.
		Filter	
		Nillable	
		Update	
Salutation	picklist	Create	Honorific added to the name for use in letters, etc.
		Filter	
		Nillable	
		Update	
• ShippingCity	string	Create	Details of the shipping address for this account. City and
ShippingCountry		Filter	country maximum size is 40 characters each, postal code and state maximum size is 20 characters each.
ShippingPostalCodeShippingState		Nillable	
		Update	
ShippingStreet	textarea	Create	The street address of the shipping address for this account. Maximum of 255 characters.
		Filter	
Field Name	Field Type	Field Properties	Description
--------------	------------	---------------------	---------------------------------------------------------------
		Nillable	
		Update	
Sic	string	Create	Standard Industrial Classification code of the company's main
		Filter	Maximum of 20 characters.
		Nillable	
		Update	
Site	string	Create	Name of the account's location, for example Headquarters
		Filter	characters.
		Nillable	
		Update	
TickerSymbol	string	Create	The stock market symbol for this account. Maximum of 20
		Filter	characters.
		Nillable	
		Update	
Туре	picklist	Create	Type of account, for example, Customer, Competitor, or
		Filter	Partner.
		Nillable	
		Update	
Website	url	Create	The website of this account. Maximum of 255 characters.
		Filter	
		Nillable	
		Update	

IsPersonAccount Fields

These fields are the subset of person account fields that are contained in the child person contact record of each person account. If the IsPersonAccount field has the value false, the following fields have a null value and cannot be modified. If true, the fields contain the value indicated in the Description column in the following table and can be modified.

Person accounts are not enabled by default. For more information, see "Person Account Fields" in the Salesforce online help.

Field Name	Field Type	Field Properties	Description
FirstName	string	Create Filter Nillable	First name of the person for a person account. Maximum size is 40 characters.

Field Name	Field Type	Field Properties	Description
		Update	
LastName	string	Create	Last name of the person for a person account. Required if
		Filter	information about person accounts, see the Salesforce online
		Nillable	help and Person Account Record Types. Maximum size is 80 characters
		Update	
PersonAssistantName	string	Create	The person account's assistant name. Label is Assistant.
		Filter	Maximum size is 40 characters.
		Nillable	
		Update	
PersonAssistantPhone	phone	Create	The person account's assistant phone. Label is Asst. Phone .
		Filter	Maximum size is 40 characters.
		Nillable	
		Update	
PersonBirthDate	date	Create	The assistant name. Label is Birthdate .
		Filter	
		Nillable	
		Update	
PersonContactId	reference	Filter	The ID for the contact associated with this person account.
		Nillable	Label 18 Contact ID.
		Update	
PersonDepartment	string	Create	The department. Label is Department . Maximum size is 80
		Filter	characters.
		Nillable	
		Update	
PersonEmail	email	Create	The person account's email. Label is Email .
		Filter	
		Nillable	
		Update	

Field Name	Field Type	Field Properties	Description
PersonEmailBouncedDate	dateTime	Create	If bounce management is activated and an email sent to the
		Filter	occurred.
		Nillable	
		Update	
PersonEmailBouncedReason	string	Create	If bounce management is activated and an email sent to the
		Filter	person account bounces, the reason the bounce occurred.
		Nillable	
		Update	
PersonHasOptedOutOfEmail	boolean	Create	Indicates whether the person account has opted out of email (true) or not (false). Label is Email Opt Out
		Defaulted	(crue) of not (rurse). Laber is Linan opt out.
		on create	
		Filter	
		Update	
PersonHomePhone	phone	Create	The person account's home phone. Label is Home Phone.
		Filter	
		Nillable	
		Update	
PersonLastCURequestDate	dateTime	Create	The last date that the person account was requested. Label is Last Stav-in-Touch Request Date
		Filter	
		Nillable	
		Update	
PersonLastCUUpdateDate	dateTime	Create	The last date a person account was updated. Label is Last Stav-in-Touch Save Date .
		Filter	
		Nillable	
		Update	
PersonLeadSource	picklist	Create	The person account's lead source. Label is Lead Source.
		Filter	
		Nillable	
		Update	
• PersonMailingCity	string	Create	Details about the person account's mailing city. Labels are Mailing City. Mailing Country. Postal Code and State
 PersonMailingCountry 		Filter	thaning City, maining Country, I ostar Couc, and State.

Field Name	Field Type	Field Properties	Description
• PersonMailingPostalCode		Nillable	Maximum size for city and country is 40 characters.
• PersonMailingState		Update	Maximum size for postal code and state is 20 characters.
PersonMailingStreet	textarea	Create	The person account's mailing street address. Label is Mailing
		Filter	Street. Maximum size is 255 characters.
		Nillable	
		Update	
PersonMobilePhone	phone	Create	The person account's mobile phone number. Label is Mobile .
		Filter	
		Nillable	
		Update	
• PersonOtherCity	string	Create	Details about the person account's alternate address. Labels
PersonOtherCountry Democratic Conde		Filter	are Other City, Other Country, Other Zip/Postal Code, and Other State.
PersonOtherState		Nillable	
		Update	
PersonOtherPhone	phone	Create	The person account's alternate phone. Label is Other Phone .
		Filter	
		Nillable	
		Update	
PersonOtherStreet	textarea	Create	The person account's alternate street address. Label is Other
		Filter	Street.
		Nillable	
		Update	
PersonTitle	string	Create	The person account's title. Label is Title . Maximum size is
		Filter	80 characters.
		Nillable	
		Update	



Note: If you are importing Account data into Salesforce and need to set the value for an audit field, such as CreatedDate, contact salesforce.com. Audit fields are automatically updated during API operations unless you request to set these fields yourself. For more information, see System Fields.

Use this object to query and manage accounts in your organization. Client applications can Create(), Update(), and Query() Attachments associated with an account via the API.

If the values in the IsPersonAccount Fields are not null, you cannot change IsPersonAccount to false, or an error will occur. For more information about person accounts, see Person Account Record Types, and the Salesforce online help.

AccountContactRole

Represents the role that a given Contact plays on an Account.

Corresponds to an SObject4 in which the ObjectType="AccountContactRole".

Supported Calls

```
Create(),Update(),Query(),Retrieve(),CreateObject(),GetDeleted(),GetUpdated(),
DescribeSObjects()
```

Special Access Rules

Customer Portal users cannot access this object.

Field	Field Type	Field Properties	Description
AccountId	reference	Create	Required. ID of the Account. For information on IDs, see
		Filter	ID Field Type.
		Update	
ContactId	reference	Create	Required. ID of the Contact associated with this account.
		Filter	For information on IDs, see ID Field Type.
		Update	
IsDeleted boo	boolean	Create	Indicates whether the object has been moved to the Recyc Bin (true) or not (false). Label is Deleted .
		Defaulted on create	
IsPrimary	boolean	Create	Specifies whether the Contact plays the primary role on the
		Defaulted on create	Account (true) or not (false). Note that each account has only one primary contact role. Label is Primary . Default value is false.
		Filter	
		Update	
Role	picklist	Create	Name of the role played by the Contact on this Account,
		Filter	such as Decision Maker, Approver, Buyer, and so on. Must be unique—there cannot be multiple records in which the

Field	Field Type	Field Properties	Description
		Nillable	AccountId, ContactId, and Role values are identical.
		Update	A contact can play different roles on the same account.

Use this object to define the role that a given Contact plays on a given Account within the context of a specific Opportunity.

AccountHistory

Represents the history of changes to the values in the fields of an account. This object is available in versions 11.0 and later.

Supported Calls

Query(),Retrieve(),GetDeleted(),GetUpdated(),DescribeSObjects()

Fields

Field	Field Type	Field Properties	Description
Field	picklist	Filter	The name of the field that was changed.
		Restricted picklist	
IsDeleted	boolean	Defaulted on create	Indicates whether the object has been moved to the Recycle Bin (true) or not (false). Label is Deleted .
		Filter	
AccountId	reference	Filter	ID of the Account. For information on IDs, see ID Field Type. Label is Account ID.

Usage

Use this object to identify changes to an account.

This object respects field level security on the parent object.

AccountOwnerSharingRule

Represents the rules for sharing an account with Users other than the owner. For more information, see Setting Sharing Rules in the Salesforce online help.



Note: Contact salesforce.com customer support to enable access to this object for your organization.

Supported Calls

```
Create(),Update(),Query(),Retrieve(),CreateObject(),GetDeleted(),GetUpdated(),
DescribeSObjects()
```

Special Access Rules

Customer Portal users cannot access this object.

Field	Field Type	Field Properties	Description
AccountAccessLevel	picklist	Create Filter Restricted picklist Update	 A value that represents the type of sharing being allowed. The possible values are: Read Edit All (This value is not valid for Create() or Update() calls.)
CaseAccessLevel	picklist	Create Filter Restricted picklist Update	 A value that represents the type of access granted to the target Group for all child cases. The possible values are: None Read Edit
ContactAccessLevel	picklist	Create Filter Restricted picklist Update	 A value that represents the type of access granted to the target Group, UserRole, or User for any associated contacts. The possible values are: None Read Edit Note: When DefaultContactAccess is set to "Controlled by Parent," you cannot create or update this field.
GroupId	reference	Create Filter	The ID representing the source group. Accounts owned by Users in the source Group trigger the rule to give access.
OpportunityAccessLevel	picklist	Create Filter Restricted picklist	A value that represents the type of access granted to the target Group for any associated Opportunity. The possible values are: • None • Read

Field	Field Type	Field Properties	Description
		Update	• Edit
UserorGroupId	reference	Create Filter	The ID representing the User or Group being granted access.

Use this object to manage the sharing rules for a particular object. General sharing and territory management-related sharing use this object.

AccountPartner

This read-only object represents a partner relationship between two Account objects. It is automatically created when a Partner object is created for a partner relationship between two accounts.



Note: This object is completely distinct and independent of Account records that have been enabled for the PRM Partner Portal.

Supported Calls

Query(),Retrieve(),DescribeSObjects()

Special Access Rules

Customer Portal users cannot access this object.

Field	Field Type	Field Properties	Description
AccountFromId	reference	Filter	ID of the main Account in the partner relationship. For information on IDs, see ID Field Type.
AccountToId	reference	Filter	ID of the partner Account in the partner relationship.
IsDeleted	boolean	Defaulted on create Filter	Indicates whether the object has been moved to the Recycle Bin (true) or not (false). Label is Deleted .
IsPrimary	boolean	Defaulted on create Filter	Indicates whether the AccountPartner is the main account's primary partner (true) or not (false).
Role	picklist	Filter Nillable	The UserRole that the partner Account has on the main Account. For example, "Consultant" or "Distributor."

Creating an Account-Account Partner Relationship

When you create a partner relationship between two accounts (when you create a Partner object and specify the AccountFromId), the API automatically creates two AccountPartner objects, one for the forward relationship and one for the reverse. For example, if you create a Partner object with "Acme, Inc." as the AccountFromId and "Acme Consulting" as the AccountToId, the API automatically creates two AccountPartner objects:

- The forward relationship AccountPartner with "Acme, Inc." as the AccountFromId and "Acme Consulting" as the AccountToId.
- The reverse relationship AccountPartner with "Acme Consulting" as the AccountFromId and "Acme, Inc." as the AccountToId.
- The value of the Role field in the reverse relationship AccountPartner is set to the PartnerRole object ReverseRole value associated with the value of the Role field in the forward relationship AccountPartner.

This mapping allows the API to manage the objects and their relationship efficiently.

AccountShare

Represents a sharing entry on an Account.

Corresponds to an SObject4 in which the ObjectType="AccountShare".

Supported Calls

Create(),Update(),Query(),Retrieve(),CreateObject(),DescribeSObjects()

Special Access Rules

Customer Portal users cannot access this object.

Field	Field Type	Field Properties	Description
AccountAccessLevel	picklist	Create Filter Restricted picklist Update	 Level of access that the User or Group has to the Account. The possible values are: Read Edit All (This value is not valid for Create () or Update () calls.) This field must be set to an access level that is at least equal to the organization's default Account access level. In addition,
			either this field, the OpportunityAccessLevel field, or the CaseAccessLevel field must be set higher than the organization's default access level.
AccountId	reference	Create Filter	ID of the Account associated with this sharing entry. This field cannot be updated. For information on IDs, see ID Field Type.

Field	Field Type	Field Properties	Description
CaseAccessLevel	picklist	Create Defaulted on create Filter Restricted picklist Update	Level of access that the User or Group has to cases associated with the account. The possible values are: • None • Read • Edit This field must be set to an access level that is at least equal to the organization's default CaseAccessLevel. This field cannot be updated via the API if the AccountAccessLevel field is set to "All." You cannot update this field for the associated account owner via the API. You must update the account owner's CaseAccessLevel via the Salesforce user interface.
ContactAccessLevel	picklist	Create Defaulted on create Filter Restricted picklist Update	Level of access that the User or Group has to contacts associated with the account. The possible values are: • None • Read • Edit This field must be set to an access level that is at least equal to the organization's default ContactAccessLevel. This field cannot be updated via the API if the ContactAccessLevel field is set to "Controlled by Parent," you cannot update this field for the associated account owner using the API. You must update the account owner's ContactAccessLevel via the Salesforce user interface.
IsDeleted	boolean	Defaulted on create Filter	Indicates whether the object has been moved to the Recycle Bin (true) or not (false). Label is Deleted .
OpportunityAccessLevel	picklist	Create Filter Restricted picklist Update	Level of access that the User or Group has to opportunities associated with the Account. The possible values are: • None • Read • Edit This field must be set to an access level that is at least equal to the organization's default opportunityAccessLevel. This field cannot be updated via the API if the AccountAccessLevel field is set to "All." You cannot use the API to update this field for the associated Account owner. You must update the Account owner's opportunityAccessLevel via the Salesforce user interface.
RowCause	picklist	Filter Restricted picklist	 Reason that this sharing entry exists. Read-only. There are many possible values, including: Manual Sharing - The User or Group has access because a User with "All" access manually shared the Account with them.

Field	Field Type	Field Properties	Description
			 Owner - The User is the owner of the Account or is in a Role above the Account owner in the role hierarchy. Sales Team—The User or Group has team access (is an AccountTeamMember). Sharing Rule—The User or Group has access via an Account sharing rule.
UserOrGroupId	reference	Create Filter	ID of the User or Group that has been given access to the Account. This field cannot be updated.

This object allows you to determine which users and groups can view and/or edit Account records owned by other users. For more information, see Sharing.

If you attempt to create an AccountShare record that matches an existing record, the Create() call updates any modified fields and returns the existing record.

AccountTag

Associates a word or short phrase with an Account.

Supported Calls

Create(),Query(),Retrieve(),DescribeSObjects()

Field	Field Type	Field Properties	Description
ItemId	reference	Create Filter	ID of the tagged item.
Name	string	Create Filter	Name of the tag. If this value does not already exist, a new TagDefinition is created and becomes the parent of this Tag object. Otherwise, a TagDefinition with the same name becomes the parent of this Tag object. Parent relationships are created automatically.
TagDefinitionId	reference	Filter	ID of the parent TagDefinition object that owns the tag.
Туре	picklist	Create Filter Restricted picklist	 Defines the visibility of a tag. Possible value are: Public: The tag can be viewed and manipulated by all users in an organization Personal: The tag can be viewed or manipulated only by a user with a matching OwnerId

Account Tag stores the relationship between its parent TagDefinition and the Account being tagged. Tag objects act as metadata, allowing users to describe and organize their data.

When a tag is deleted, its parent TagDefinition will also be deleted if the name is not being used; otherwise, the parent remains. Deleting a TagDefinition sends it to the recycle bin, along with any associated tag entries.

For more information on tags, see "About Tagging" in the Salesforce online help.

AccountTeamMember

Represents a User who is a member of an Account team. See also UserAccountTeamMember, which represents a User who is on the default account team of another user.

Corresponds to an SObject4 in which the ObjectType="AccountTeamMember".

Supported Calls

```
Create(),Update(),DescribeSObjects()Query(),Search(),Retrieve(),CreateObject(),
GetDeleted(),GetUpdated()
```

Special Access Rules

- This object is available only for Enterprise and Unlimited Edition users who have enabled the account team functionality.
- Customer Portal users cannot access this object.

Field	Field Type	Field Properties	Description
AccountAccessLevel	picklist	Filter Nillable Restricted picklist	Level of access that the User or Group has to the Account. The possible values are: None Read Edit All This field must be set to an access level that is at least equal to the organization's default Account access level. In addition, the AccountAccountAccessLevel, OpportunityAccessLevel, or CaseAccessLevel field must be set higher than the organization's default access level.
AccountId		Create Filter	Required. ID of the Account to which this user is a team member. Must be a valid account ID. Required. For information on IDs, see ID Field Type.
IsDeleted	boolean	Defaulted on create Filter	Indicates whether the object has been moved to the Recycle Bin (true) or not (false). Label is Deleted .

Field	Field Type	Field Properties	Description
TeamMemberRole	picklist	Create Role associated with this team member. On	Role associated with this team member. One of the valid
		Filter	team member roles defined for your organization. Label is Team Role
		Nillable	
		Update	
UserId	reference	Create	Required. ID of the User who is a member of this account
		Filter	team. Must be a valid User ID. Required.

Use this object to manage the team members of a particular Account and to specify team member roles for those users on that account.

AccountTerritoryAssignmentRule

An account assignment rule that assigns accounts to territories based on account fields. Only available if territory management has been enabled for your organization. For more information, see the "What is Territory Management?" topic in the Salesforce online help.

Supported Calls

Create(),Update(),Query(),Retrieve(),DescribeSObjects(),GetDeleted(),GetUpdated()

Special Access Rules

Customer Portal users cannot access this object.

Field	Field Type	Field Properties	Description
BooleanFilter	string	Create Filter Nillable Update	Advanced filter conditions that were specified for the rule in the online application. For example, "(1 AND 2) OR 3."
IsActive	boolean	Create Defaulted on create Filter Update	Indicates whether the rule is active (true) or inactive (false). Via the API, active rules run automatically when new accounts are created and existing accounts are edited. The exception is when the IsExcludedFromRealign field on an account is true, which prevents account assignment rules from evaluating that account.

Field	Field Type	Field Properties	Description
IsInherited	boolean	Create Defaulted on create Filter Update	Indicates whether the rule is an inherited rule (true) or a local rule (false). An inherited rule also acts upon territories below it in the territory hierarchy. A local rule is created at the immediate territory and only impacts the immediate territory.
Name	string	Create Filter Update	A name for the rule. Limit is 80 characters.
TerritoryId	reference	Create Filter Update	ID of the territory where accounts that satisfy this rule are assigned. For information on IDs, see ID Field Type.

A territory will not have any accounts (with the exception of manually assigned accounts) unless at least one account assignment rule is active for the territory.

AccountTerritoryAssignmentRuleItem

A row of selection criteria for an Account Territory Assignment Rule. Only available if territory management has been enabled for your organization. For more information, see the "What is Territory Management?" topic in the Salesforce online help.

Supported Calls

```
Create(),Update(),Query(),Retrieve(),DescribeSObjects(),GetDeleted(),GetUpdated()
```

Special Access Rules

Customer Portal users cannot access this object.

Field	Field Type	Field Properties	Description
Field	picklist	Create	The standard or custom account field to use as a criteria.
		Filter	
		Restricted picklist	
		Update	

Field	Field Type	Field Properties	Description
Operation	picklist	Create	The criteria to apply, such as "equals" or "starts with."
		Filter	
		Restricted picklist	
		Update	
RuleID	reference	Create	ID of the associated AccountTerritoryAssignmentRule.
		Filter	
		Update	
SortOrder in	int	Create	The order in which this row is evaluated compared to other
		Filter	Account lerritoryAssignmentRuleItem objects for the given AccountTerritoryAssignmentRule.
		Update	2 8
Value	string	Create	The field value(s) to evaluate, such as "94105" if the Field
		Filter	is "Billing Zip/Postal Code."
		Nillable	
		Update	

- Both standard and custom account fields can be used as criteria for account assignment rules.
- A territory will not have any accounts (with the exception of manually assigned accounts) unless at least one account assignment rule is active for the territory.

AccountTerritorySharingRule

Represents the rules for sharing an Account within a Territory.

Supported Calls

Create(),Update(),DescribeSObjects(),Query(),Retrieve(),GetDeleted(),GetUpdated()

Special Access Rules

Customer Portal users cannot access this object.

Fields

Field Type	Field Properties	Description
picklist	Create	A value that represents the type of sharing being allowed.
	Filter	The possible values are:
	Restricted picklist	 Read Edit All
	Update	
picklist	Create	A value that represents the type of access granted to the target
	Filter	group for all child cases of the account. The possible values are:
	Restricted	• None
	picklist	• Read
	Update	• Edit
picklist	Create	A value that represents the type of access granted to the target
	Filter	group for all related contacts on the account. The possible values are:
	Restricted	None
	picklist	• Read
	Update	• Edit
		Note: This field is read only.
reference	Create	The ID representing the source group. Accounts owned by
	Filter	users in the source territory trigger the rule to give access.
picklist	Create	A value that represents the type of access granted to the target
	Filter	group for all opportunities associated with the account. I ne possible values are:
	Restricted picklist	• None • Read
	Update	• Edit
reference	Create Filter	The ID representing the user or group being given access, or, if a territory ID, the users assigned to that territory.
	Field picklist picklist picklist picklist picklist reference picklist reference reference reference reference	Field PropertiesPicklistCreateFilterRestricted picklistUpdateVidatePicklistCreateFilterRestricted picklistDicklistCreateFilterRestricted picklistDicklistUpdatePicklistCreatePicklistCreateFilterRestricted picklistDicklistUpdatePicklistCreateFilterFilterPicklistUpdatePicklistUpdatePicklistUpdateVupdateUpdatePicklistUpdatePicklistUpdatePicklistUpdatePicklistUpdatePicklistCreateFilterFilterPicklistCreateFilterFilterPicklistUpdatePicklistUpdatePicklistFilterPicklistUpdatePicklistUpdatePicklistUpdatePicklistUpdatePicklistUpdatePicklistUpdatePicklistUpdatePicklistUpdatePicklistUpdatePicklistUpdatePicklistUpdatePicklistUpdatePicklistUpdatePicklistUpdatePicklistUpdatePicklistUpdatePicklistUpdatePicklistUpdatePicklist<

Usage

Use this object to manage the sharing rules for a particular object. General sharing and Territory-related sharing use this object.

ActivityHistory

This read-only object contains information about each activity related to an object.

Supported Calls

DescribeGlobal(),DescribeSObject(),andDescribeSObjects()

Field	Field Type	Field Properties	Description
ActivityDate d	date F	Filter	Indicates one of the following:
		Nillable	 For a task, the due date of a task. For an event, indicates the due date of the event only if IsAllDayEvent is set to true.
			This field has a timestamp that is always set to midnight in the Coordinated Universal Time (UTC) time zone. The timestamp is not relevant; do not attempt to alter it in order to accommodate time zone differences.
ActivityType	picklist	Filter Nillable	One of the following values: Call, Meeting, or Other.
CallDisposition	string	Filter Nillable	Represents the result of a given call, for example, "we'll call back," or "call unsuccessful." Limit is 255 characters.
CallDurationInSeconds	int	Filter Nillable	Duration of the call in seconds.
CallObject	string	Filter Nillable	Name of a call center. Limit is 255 characters.
CallType	picklist	Filter Nillable	The type of call being answered: Inbound, Internal, or Outbound.
Description	textarea	Nillable	Description of the task or event.
DurationInMinutes	int	Filter Nillable	Length of the event or task.
IsAllDayEvent	boolean	Defaulted on create Filter	If true, the activity is an event and the ActivityDate is used to define the date of the event. If false, the activity may be a task or an event.

Field	Field Type	Field Properties	Description
IsClosed	boolean	Defaulted on create Filter	For tasks only, indicates whether the task was completed (true) or not (false). This field is set indirectly by setting the Status fieldeach picklist value has a corresponding IsClosed value.
IsDeleted	boolean	Defaulted on create Filter	Indicates whether the object has been moved to the Recycle Bin (true) or not (false). Label is Deleted .
IsOnlineMeeting	boolean	Defaulted on create Filter	Indicated whether the activity represents an online meeting (true) or not (false).
IsTask	boolean	Defaulted on create Filter	If true, the activity is a task. If false, it is an event.
IsVisibleInSelfService	boolean	Defaulted on create Filter	If true, the activity can be viewed in the self-service portal.
Location	string	Filter Nillable	If an event, the location of the event. If not, the value is null.
OwnerId	reference	Filter Nillable	ID of the user who owns the task or event.
Priority	picklist	Filter Nillable	If a task, the importance of the task, such as high, normal, or low.
Status	picklist	Filter Nillable	For tasks, the current status of the task, for example In Progress or Complete. Each predefined Status field sets a value for Isclosed. To obtain picklist values, you can Query () on the TaskStatus object.
Subject	comobobox	Filter Nillable	Subject line of the task or event.
WhatId	reference	Filter Nillable	ID of the related object (Account, Campaign, Case, Opportunity, or custom object).
WhoId	reference	Filter Nillable	ID of the related Contact or Lead. If WhoId refers to a lead, then the WhatId field must be empty.

This object helps you replicate the related list functionality of the Salesforce user interface. To use this object, use the following procedure:

- 1. Optionally, issue a describe call against the object whose activities you wish to query, to get a suggestion of the correct SOQL to use.
- 2. Issue a SOQL relationship query with a main clause that references the object, and an inner clause that references the activity history, for example:

```
SELECT (SELECT ActivityDate, Description from ActivityHistories) FROM Account WHERE Name Like 'XYZ%'
```

or

```
SELECT (SELECT ActivityDate, Description from OpenActivities) FROM Account WHERE Name Like 'XYZ%'
```

The Salesforce user interface enforces sharing rules, filtering out related list items that a user does not have permission to see.

In order to prevent performance issues while still providing the related list functionality, there are some restrictions on users who do not have "View All Data" permission. Such users must comply with the following restrictions:

- In the main clause of the relationship query, you can reference only one record. For example, you cannot filter on all records where the account name starts with A, but must reference a single account record.
- You cannot use WHERE clauses.
- You must specify a limit to the number of rows returned, less than 500.
- You must sort on ActivityDate and LastModifiedDate, descending order: ORDER BY ActivityDate DESC, LastModifiedDate DESC

You cannot use Query () directly on this object.

AdditionalNumber

This object represents an optional additional number for a call center. This additional number is visible in the call center's phone directory.

Supported Calls

```
Create(),Query(),Retrieve(),GetDeleted(),GetUpdated(),DescribeSObject(),
DescribeSObjects()
```

Special Access Rules

Customer Portal users cannot access this object.

Field	Field Type	Field Properties	Description
CallCenterId	reference	Create Filter Nillable Update	System field that contains the ID of the user who created the call center associated with this additional number. If value is null, this additional number is displayed in every call center's phone directory.
CreatedById	reference	Defaulted on create Filter	System field that contains the ID of the user who created the call center associated with this additional number.
CreatedDate	dateTime	Defaulted on create Filter	System field that contains the date and time that the call center associated with this additional number was created.
Description	string	Create Filter Nillable Update	Description of the additional number, such as Conference Room B. Limit: 255 characters.
Id	ID	Defaulted on create Filter	ID that uniquely identifies this call center. Label is Additional Directory Number ID.
IsDeleted	boolean	Create Filter Nillable	Indicates whether the additional number has been deleted (true) or not (false). Label is Deleted .
LastModifiedById	reference	Defaulted on create Filter	System field that contains the ID of the user who last modified this additional number.
LastModifiedDate	dateTime	Defaulted on create Filter	System field that contains the date and time this additional number was last modified.
Name	string	Create Filter	The name of the additional number. Limit: 80 characters.
Phone	phone	Create Filter Update	The phone number that corresponds to this additional number.

Field	Field Type	Field Properties	Description
SystemModstamp	dateTime	Defaulted on create Filter	System field that contains the date and time that this additional number was last modified by a user or a workflow process.

Create an additional number for a call center directory. Use this object if the number is not easily categorized as a User, Contact, Lead, Account, or the other object. Examples include phone queues or conference rooms.

ApexClass

Represents an Apex class. For information, see the Apex Language Reference.

Corresponds to an SObject4 in which the ObjectType="ApexClass".



Note: Although Apex classes and triggers have the Create and Update field properties set to true, a runtime exception occurs if you try to create or update them using the API. Therefore, use the Force.com Migration Tool, the Salesforce user interface, or the Force.com IDE to create or update Apex classes or triggers.

Supported Calls

```
Create(),Update(),Query(),Retrieve(),CreateObject(),DescribeSObjects(),GetDeleted(),
GetUpdated()
```

Field	Field Type	Field Properties	Description
ApiVersion	double	Create	The API version for this class. Every class has an API version
		Filter	specified at creation.
		Nillable	
		Update	
Body	textarea	Create	The Apex class definition.
		Nillable	Limit: 100,000 characters.
		Update	
IsValid	boolean	Create	Indicates whether any dependent metadata has changed since
		Defaulted on create	the class was last compiled (true) or not (false).
		Update	

Field	Field Type	Field Properties	Description
LengthWithoutComments	int	Create Filter Update	Length of the class without comments.
Name	string	Create Filter Update	Name of the class. Limit: 255 characters
NamespacePrefix	string	Create Filter Nillable	The namespace prefix assigned to this object when it was created as part of a managed package. Null if this object is not part of a managed package. Limit is 15 characters. For more information about managed packages and namespace prefixes, see "About Managed Packages" in the Salesforce online help.
Status	picklist	Create Filter Restricted picklist Update	 The current status of the Apex class. The following string values are valid: Active The class is active. Deleted The class is marked for deletion. This is useful for managed packages, because it allows a class to be deleted when a managed package is updated. Note: The ApexTrigger Status field includes an Inactive option, but it is only supported for ApexTrigger. For more information, see the Force.com Metadata API Developer's Guide.

For more information about Apex classes, see the Apex Language Reference.

ApexComponent

Represents a definition for a custom component that can be used in a Visualforce page alongside standard components such as <apex:relatedList> and <apex:dataTable>. For information, see the *Visualforce Developers Guide* at www.salesforce.com/us/developer/docs/pages/index.htm.

Corresponds to an SObject4 in which the ObjectType="ApexComponent".

Supported Calls

```
Create(),Update(),Query(),Retrieve(),CreateObject(),DescribeSObjects(),GetDeleted(),
GetUpdated()
```

Field	Field Type	Field Properties	Description
ControllerKey	string	Create Filter Nillable Update	 The identifier for the controller associated with this custom component: If the ControllerType parameter is set to Standard or StandardSet, this value is the name of the sObject that defines the controller. If the ControllerType parameter is set to Custom, this value is the name of the Apex class that defines the controller.
ControllerType	picklist	Create Filter Update	 The type of controller associated with this Visualforce custom component. Possible values include: Not Specified, for custom components defined without a value for the controller attribute on the <apex:component> tag</apex:component> Standard, a value that cannot be used with custom components or errors may occur StandardSet, a value that cannot be used with custom components or errors may occur Custom, for components that have a value for the controller attribute on the <apex:component> tag</apex:component>
Description	string	Create Filter Nillable Update	Description of the Visualforce custom component.
Markup	textarea	Create Update	The Visualforce markup, HTML, Javascript, and any other Web-enabled code that defines the content of the custom component.
MasterLabel	string	Create Filter Update	The text used to identify the Visualforce custom component in the Setup area of Salesforce. The Label for this field is Label.
Name	string	Create Filter Update	Required. Name of this Visualforce custom component.
NamespacePrefix	string	Create Filter Nillable	The namespace prefix assigned to this object when it was created as part of a managed package. Null if this object is not part of a managed package. Limit is 15 characters. For more information about managed packages and namespace prefixes, see "About Managed Packages" in the Salesforce online help.

Use custom components to encapsulate a common design pattern and then reuse that pattern several times in one or more Visualforce pages. All users who can view Visualforce pages can view custom components, but the "Customize Application" permission is required to Create() or Update() custom components.

ApexPage

Represents a single Visualforce page. For information, see the *Visualforce Developers Guide* at www.salesforce.com/us/developer/docs/pages/index.htm.

Corresponds to an SObject4 in which the ObjectType="ApexPage".

Supported Calls

```
Create(),Update(),Query(),Retrieve(),CreateObject(),DescribeSObjects(),GetDeleted(),
GetUpdated()
```

Field	Field Type	Field Properties	Description
ControllerKey	string	Create Filter Nillable Update	 The identifier for the controller associated with this page: If the ControllerType parameter is set to Standard or StandardSet, this value is the name of the sObject that defines the controller. If the ControllerType parameter is set to Custom, this value is the name of the Apex class that defines the controller.
ControllerType	picklist	Create Filter Update	 The type of controller associated with this Visualforce page. Possible values include: Not Specified, for pages defined with neither a standardController nor a controller attribute on the <apex:page> tag</apex:page> Standard, for pages defined with the standardController attribute on the <apex:page> tag</apex:page> StandardSet, for pages defined using the standardController and recordSetVar attribute on the <apex:page> tag</apex:page> Custom, for pages defined with the controller attribute on the <apex:page> tag</apex:page>
Description	string	Create Filter Nillable Update	Description of the Visualforce page.

Field	Field Type	Field Properties	Description
Markup	textarea	Create Update	The Visualforce markup, HTML, Javascript, and any other Web-enabled code that defines the content of the page.
MasterLabel	string	Create Filter Update	The text used to identify the Visualforce page in the Setup area of Salesforce. The Label is Label .
Name	string	Create Filter Update	Required. Name of this Visualforce page.
NamespacePrefix	string	Create Filter Nillable	The namespace prefix assigned to this object when it was created as part of a managed package. Null if this object is not part of a managed package. Limit is 15 characters. For more information about managed packages and namespace prefixes, see "About Managed Packages" in the Salesforce online help.

Use Visualforce pages to add custom content that extends the base Salesforce application functionality. All users in Visualforce-enabled organizations can view Visualforce pages, but the "Customize Application" permission is required to Create() or Update() them.

ApexTrigger

Represents an Apex trigger.

Corresponds to an SObject4 in which the ObjectType="ApexClass".



Note: Although Apex classes and triggers have the Create and Update field properties set to true, a runtime exception occurs if you try to create or update them using the API. Therefore, use the Force.com Migration Tool, the Salesforce user interface, or the Force.com IDE to create or update Apex classes or triggers.

Supported Calls

```
Create(),Update(),Query(),Retrieve(),CreateObject(),DescribeSObjects(),GetDeleted(),
GetUpdated()
```

Field	Field Type	Field Properties	Description
ApiVersion	double	Create	The API version for this trigger. Every trigger has an API
		Filter	version specified at creation.

Field	Field Type	Field Properties	Description
		Update	
Body	textarea	Create	The Apex trigger definition.
		Nillable	Limit: 32,000 characters.
		Update	
IsValid	boolean	Create	Indicates whether any dependent metadata has changed since
		Defaulted on create	the trigger was last compiled (true) or not (false).
		Update	
LengthWithoutComments	int	Create	Length of the trigger without comments
		Filter	
		Update	
Name	string	Create	Name of the trigger.
		Filter	Limit: 255 characters
		Update	
NamespacePrefix	string	Create	The namespace prefix assigned to this object when it was
		Filter	part of a managed package. Limit is 15 characters. For more
		Nillable	information about managed packages and namespace prefixes, see "About Managed Packages" in the Salesforce online help.
Status	picklist	Create	The current status of the Apex trigger. The following string
		Filter	values are valid: • Active The trigger is active
		Restricted picklist	 Inactive The trigger is inactive, but not deleted. Deleted The trigger is marked for deletion. This is
		Update	useful for managed packages, because it allows a class to be deleted when a managed package is updated.
			Note: Inactive is not valid for ApexClass. For more information, see the <i>Force.com Metadata API Developer's Guide</i> .

For more information about Apex triggers, see the Apex Language Reference.

Approval

Represents an approval request for a Contract.

Corresponds to an SObject4 in which the ObjectType="Approval".



Note: This object exists for backwards compatibility. This object is specific to approvals on the Contract object. It is not equal to or involved in the approval processes represented by the ProcessInstance object. ProcessInstances are more powerful.

Supported Calls

```
Create(),Update(),Query(),Search(),Retrieve(),CreateObject(),GetUpdated(),
DescribeSObjects()
```

Special Access Rules

Customer Portal users cannot access this object.

Field	Field Type	Field Properties	Description
ApproveComment	string	Create	Text entered by the user when they approved or rejected this
		Query	approval request. Required. Limit: 4,000 characters.
		Replicate	
		Retrieve	
		Update	
IsDeleted	boolean	Defaulted on create	Indicates whether the object has been moved to the Recycle Bin (true) or not (false). Label is Deleted .
		Filter	
OwnerId	reference	Create	Required. ID of the User being asked to approve or reject
		Filter	the approval request. Must be a valid User ID. Required. For information on IDs, see ID Field Type.
		Update	
ParentId	reference	Create	Required. ID of the Contract associated with this approval
		Filter	request. Must be a valid contract ID.
RequestComment	string	Create	Text entered by the User who created the approval request.
		Filter	has been created. Limit: 4,000 characters.
		Nillable	
		Update	
Status	picklist	Create	Required. Status of this approval request. One of the
		Filter	 following picklist values: Pending—Specified only when the Approval request
		Restricted	is created (Create () call)
		pickiist	• Approved—Specified only when the Approval request is approved (Update () call)
		Opdate	is approved (Opdate () call)

Field	Field Type	Field Properties	Description
			• Rejected—Specified when the Approval request is rejected (Update() call) or when it is created (Create() call) and immediately rejected for archival/historical purposes.

This object allows client applications to programmatically handle approval requests for a Contract. Initially, to request a Contract approval, a client application might create a new Approval request record, specifying the ParentId, OwnerId (user approving or rejecting the request), Status (Pending), and (optionally) RequestComment fields. Note that when a client application creates the first approval request, if the value of the ContractStatus field is Draft, then the Status for this record is automatically changed to InApproval (see ContractStatus for more information).

A client application might subsequently update an existing Approval request, specifying the Status (Approved or Rejected) and an ApproveComment (required); the RequestComment field cannot be updated. Updating an Approval record (either to approve or reject) requires the client application to be logged in with "Approve Contract" permission. To update an Approval request, its Status must be Pending—a client application cannot update an Approval that has already been Approved or Rejected. To re-submit an approval request for a given Contract, a client application must create a new, separate Approval record and repeat the approval process.

Once a Contract has been approved (not rejected), the ContractLastApprovedDate field is automatically updated, however the ContractStatus field is not updated, it keeps the value InApproval.

An approved Contract must be activated explicitly. Client applications can activate a Contract by setting the value in its Status field to Activated, or Users can activate a Contract via the Salesforce user interface.

A Contract can have multiple approval requests in various states (Pending, Approved, and Rejected). In addition, one User can have multiple approval requests associated with the same Contract.

Client applications cannot explicitly delete Approval records. Approval records are deleted automatically if the parent Contract is deleted.

Asset

Represents an item of commercial value owned by an Account or Contact, for example, a product previously sold and installed.

Corresponds to an SObject4 in which the ObjectType="Asset".

Supported Calls

```
Create(),Update(),Query(),Search(),Retrieve(),CreateObject(),GetDeleted(),GetUpdated(),
DescribeSObjects()
```

Field	Field Type	Field Properties	Description
AccountId	reference	Create	Required. ID of the Account associated with this asset. Must
		Filter	be a valid account ID. Required if ContactId is not specified. For information on IDs, seeID Field Type.
		Nillable	, , , , , , , , , , , , , , , , , , ,
		Update	
ContactId	reference	Create	Required if AccountId is not specified. ID of the Contact
		Filter	associated with this asset. Must be a valid contact ID that
		Nillable	AccountId).
		Update	
Description	string	Create	Description of this asset.
		Nillable	
		Update	
InstallDate	date	Create	Date on which this asset was installed.
		Filter	
		Nillable	
		Update	
IsCompetitorProduct	boolean	Create	Indicates whether this Asset represents a product sold by a
		Defaulted on create	competitor (true) or not (false). Default value is false. Label is Competitor Asset .
		Filter	
		Update	
IsDelete	boolean	Defaulted on create	Indicates whether the object has been moved to the Recycle Bin (true) or not (false). Label is Deleted .
		Filter	
Name	string	Create	Required. Name of the asset. Label is Asset Name .
	Ŭ	Filter	
		Update	
Price	double	Create	Price paid for this asset.
		Filter	
		Nillable	
		Update	
Product2Id	reference	Create	ID of the Product2 associated with this asset. Must be a valid
		Filter	Product2 ID. Optional.

Field	Field Type	Field Properties	Description
		Nillable	
		Update	
PurchaseDate	date	Create	Date on which this asset was purchased.
		Filter	
		Nillable	
		Update	
Quantity	double	Create	Quantity purchased or installed.
		Filter	
		Nillable	
		Update	
SerialNumber	string	Create	Serial number for this asset.
		Filter	
		Nillable	
		Update	
Status	picklist	Create	Customizable picklist of values. The default picklist includes
		Filter	Purchased
		Nillable	• Shipped
		Update	• Installed
			Kegistered Obsolete
UsageEndDate	date	Create	Date when usage for this asset ends or expires.
		Filter	
		Nillable	
		Update	

Use this object to track assets previously sold into customer accounts. With asset tracking, a client application can quickly determine which products were previously sold or are currently installed at a specific account.

For example, your organization might want to renew and up-sell opportunities on products sold in the past. Similarly, your organization might want to track competitive products that exist in a customer environment that could potentially be replaced or swapped out.

Asset tracking is also useful for product support, providing detailed information to assist with product-specific support issues. For example, the PurchaseDate or SerialNumber could indicate whether a given product has certain maintenance requirements, including product recalls. Similarly, the UsageEndDate might indicate when the asset was removed from service or when a license or warranty expires.

If an application creates a new Asset record, it must at least specify a Name and either an AccountId, ContactId, or both.

AssetTag

Associates a word or short phrase with an Asset.

Supported Calls

Create(),Query(),Retrieve(),DescribeSObjects()

Fields

Field	Field Type	Field Properties	Description
ItemId	reference	Create Filter	ID of the tagged item.
Name	string	Create Filter	Name of the tag. If this value does not already exist, a new TagDefinition is created and becomes the parent of this Tag object. Otherwise, a TagDefinition with the same name becomes the parent of this Tag object. Parent relationships are created automatically.
TagDefinitionId	reference	Filter	ID of the parent TagDefinition object that owns the tag.
Туре	picklist	Create Filter Restricted picklist	 Defines the visibility of a tag. Possible value are: Public: The tag can be viewed and manipulated by all users in an organization Personal: The tag can be viewed or manipulated only by a user with a matching OwnerId

Usage

AssetTag stores the relationship between its parent TagDefinition and the Asset being tagged. Tag objects act as metadata, allowing users to describe and organize their data.

When a tag is deleted, its parent TagDefinition will also be deleted if the name is not being used; otherwise, the parent remains. Deleting a TagDefinition sends it to the recycle bin, along with any associated tag entries.

For more information on tags, see "About Tagging" in the Salesforce online help.

AssignmentRule

Represents an assignment rule associated with a Case or Lead.

Corresponds to an SObject4 in which the ObjectType="AssignmentRule".

Supported Calls

```
Query(),Retrieve(),CreateObject(),DescribeSObjects()
```

Special Access Rules

- This object is read only. Assignment rules are created, configured, and deleted in the Salesforce user interface.
- Customer Portal users cannot access this object.

Fields

Field	Field Type	Field Properties	Description
Active	boolean	Defaulted on create Filter	Indicates whether this assignment rule is active (true) or not (false).
Name	string	Filter Nillable	Name of this assignment rule.
SobjectType	picklist	Filter Nillable Restricted picklist	Type of assignment rule—Case or Lead.

Usage

Before creating or updating a new Case or Lead, a client application can Query() (by name) the AssignmentRule to obtain the ID of the assignment rule to use, and then assigned that ID to the assignmentRuleId field of the SOAP header (see SetSOAPHeader()).

AsyncApexJob

Represents an individual Apex sharing recalculation job or method with the future annotation.

Supported Calls

Query(),Retrieve()

Field Name	Field Type	Field Properties	Description
ApexClassID	reference	Filter	The ID of the Apex class executing the job. Label is $\tt Class ID.$

Field Name	Field Type	Field Properties	Description
CompletedDate	dateTime	Filter	The date and time when the job was completed.
		Nillable	
JobItemsProcessed	int	Filter	Number of job items processed. Label is Batches Processed.
ЈорТуре	picklist	Filter	The type of job being processed. Values are future and
		Restricted picklist	SharingRecalculation.
MethodName	string	Filter	The name of the Apex method being executed. Label is Apex
		Nillable	Method.
NumberOfErrors	int	Filter	Total number of batches with a failure. A batch is consider
		Nillable	entire failure of the batch. Label is Failures.
Status	picklist	Filter	The status of the job. Valid values are:
		Restricted	• Queued
		picklist	Aborted
			• Completed
			• Failed
TotalJobItemsintFilter NillableTotal number of batches processed. set of records. Label is Total Bate	Total number of batches processed. Each batch contains a		
	Nillable	set of records. Label is Total Batches.	

Use this object to query Apex batch jobs in your organization.

Attachment

Represents a file that a User has uploaded and attached to a parent object.

Corresponds to an SObject4 in which the ObjectType="Attachment".

Supported Calls

```
Create(),DescribeSObjects(),Query(),Search(),Retrieve(),CreateObject(),
GetDeleted()GetUpdated()
```

Field	Field Type	Field Properties	Description
Body	base64	Create	Required. Encoded file data.
		Update	Note: Client applications are responsible for the conversion of Base64 data between binary and String formats.
BodyLength	int	Filter	Size of the file (in bytes).
		Nillable	
ContentType	string	Create	The content type of the attachment.
		Filter	If the Disallow HTML documents and attachments
		Nillable	upload files with the following file extensions: htm, html,
		Update	htt, htx, mhtm, mhtml, shtm, shtml, acgi.
IsDeleted	boolean	Defaulted on create	Indicates whether the object has been moved to the Recycle Bin (true) or not (false). Label is Deleted .
		Filter	
IsPrivate	boolean	Create	Indicates whether this record is viewable only by the owner
		Defaulted on create	and administrators (true) or viewable by all otherwise-allowed users (false). During a Create () or Update () call it is possible to mark an Attrachment record
		Filter	as private even if you are not the owner. This can result in a
		Update	situation in which you can no longer access the record th you just inserted or updated. Label is Private .
Name	string	Create	Required. Name of the attached file. Label is File Name .
		Filter	
		Update	
OwnerId	reference	Create	ID of the User who owns the attachment. For information
		Defaulted on create	on IDs, see ID Field Type. This field was required previous to release 9.0. Beginning with release 9.0, it can be null on create.
		Filter	
		Update	
ParentId	reference	Create	Required. ID of the parent object of the attachment. Th
		Filter	following objects are supported as parents of attachments:
			Asset
			• Campaign
			Case Contract
			Contact Contract

Field	Field Type	Field Properties	Description
			 Custom objects EmailMessage EmailTemplate Event Lead Opportunity Product2 Solution Task For information on IDs, see ID Field Type.

The API sends and receives the binary file attachment data encoded as a base64Binary data type. Prior to Create(), client applications must encode the binary attachment data as base64. Upon receiving a response, client applications must decode the base64 data to binary.

The Create () call restricts these files to a maximum size of 5 MB. For a file attached to a Solution, the limit is 1.5MB. The maximum email attachment size is 3 MB.

The API supports attachments on email in Create () and Update () calls. The Query () call does not return attachments parented by email, unless the user performing the query has the "Modify All Data" permission.

Note: The Search () call does not search Attachment records during text searches.

Access to fields depends on the method being used:

- All of the fields are accessible using the DescribeSObjects () and Query () calls. With the Create () call, you can insert the Name, ParentId, Body, IsPrivate, and OwnerId fields.
- To modify existing records, the Update() call gives you access to change the Name, Body, IsPrivate, and OwnerId fields.
- You can access all of the fields using a Query () call. However, you cannot receive the Body field for multiple records in a single Query () call. If your query returns the Body field, your client application must ensure that only one row with one Attachment is returned; otherwise, an error occurs. A more effective approach is to return IDs (but not Attachments in the Body field) from a Query () call and then pass them into Retrieve () calls that return the Body field.
- · For information about accessing the attachments of archived activities, see Archived Activities.

Bookmark

Represents a link between opportunities that share common information.

This object is available to organizations with the Similar Opportunities feature enabled.

Supported Calls

```
Query(),GetDeleted(),GetUpdated()
```

Fields

Field	Field Type	Field Properties	Description
ID	ID	Defaulted on create Filter	ID of the bookmark. Label is Bookmark ID . For information on IDs, see ID Field Type.
FromId	ID	Filter	The originating opportunity. Label is Bookmarked From ID .
Told	ID	Filter	The opportunity to which the originating opportunity is linked. Label is Bookmarked To ID .
IsDeleted	boolean	Defaulted on create Filter	Indicates whether the object has been moved to the Recycle Bin (true) or not (false). Label is Deleted .

Usage

The Bookmark object works with the Opportunity object only.

Use this read-only object to query the bookmarks between opportunities in your organization. In the online application, users can search for opportunities that share attributes with their opportunity. The user can then bookmark the appropriate opportunities for future reference.

BrandTemplate

Letterhead for HTML EmailTemplate.

Supported Calls

```
Create(),Query(),Retrieve(),Update(),GetDeleted(),GetUpdated()
```

Special Access Rules

Customer Portal users cannot access this object.

Field	Field Type	Field Properties	Description
Description	string	Create	Description of the letterhead. Limited to 1000 characters.
		Filter	
		Nillable	
		Update	
Field	Field Type	Field Properties	Description
-----------------	------------	------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
DeveloperName	string	Create Filter Nillable Update	The unique name of the object in the API. The name can contain only alphanumeric characters and must begin with a letter. 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 Letterhead Unique Name .
IsActive	boolean	Create Defaulted on create Filter Update	Indicates whether the letterhead is available for use (true) or not (false). Label is Active .
Name	string	Create Filter Update	Label of the template as it appears in the Salesforceuser interface. Limited to 255 characters. Label is Letterhead Label .
NamespacePrefix	string	Create Filter Nillable	The namespace prefix assigned to this object when it was created as part of a managed package. Null if this object is not part of a managed package. Limit is 15 characters. For more information about managed packages and namespace prefixes, see "About Managed Packages" in the Salesforce online help. This field cannot be accessed unless the logged-in user has the "Customize Application" permission.
Value	textarea	Create Update	The contents of the letterhead, in HTML, including any logos.

Use this object to brand EmailTemplates with your letterhead. You can also set a brand template to active or inactive. For example, if you have five different marketing brands, you can maintain each different brand in one template, and assign to the appropriate EmailTemplate.

BusinessHours

This object is used to specify the business hours of your support organization. Escalation rules are run only during these hours. If business hours are associated with any holidays, then business hours and escalation rules associated with business hours are suspended during the dates and times specified as holidays. For more information, see Holiday.

```
Create(),Update(),Query(),Retrieve(),GetUpdated(),CreateObject()
```

Special Access Rules

Customer Portal users cannot access this object.

Field	Field Type	Field Properties	Description
IsActive	boolean	Create	Indicates whether the business hours is active (true) or not
		Defaulted on create	active (false).
		Filter	
		Update	
Name	string	Create	The name of the business hours.
		Filter	
		Update	
IsDefault	boolean	Create	Indicates whether the business hours are set as the default
		Defaulted on create	business hours (true) or not (false).
		Filter	
		Update	
FridayEndTime	time	Create	Time that business closes.
		Filter	
		Nillable	
		Update	
FridayStartTime	time	Create	Time that business opens.
		Filter	
		Nillable	
		Update	
MondayEndTime	time	Create	Time that business closes.
		Filter	
		Nillable	
		Update	
MondayStartTime	time	Create	Time that business opens.
		Filter	
		Nillable	
		Update	

Field	Field Type	Field Properties	Description
SaturdayEndTime	time	Create	Time that business closes.
		Filter	
		Nillable	
		Update	
SaturdayStartTime	time	Create	Time that business opens.
		Filter	
		Nillable	
		Update	
SundayEndTime	time	Create	Time that business closes.
		Filter	
		Nillable	
		Update	
SundayStartTime	time	Create	Time that business opens.
		Filter	
		Nillable	
		Update	
ThursdayEndTime	time	Create	Time that business closes.
		Filter	
		Nillable	
		Update	
ThursdayStartTime	time	Create	Time that business opens.
		Filter	
		Nillable	
		Update	
TimeZoneSidKey	picklist	Create	The time zone of the business hours.
		Filter Restricted picklist Update	
TuesdayEndTime	time	Create	Time that business closes.
		Filter	
		Nillable	
		Update	

Field	Field Type	Field Properties	Description
TuesdayStartTime	time	Create Filter	Time that business opens.
		Nillable Update	
WednesdayEndTime	time	Create Filter Nillable Update	Time that business closes.
WednesdayStartTime	time	Create Filter Nillable Update	Time that business opens.

Use this object to specify the business hours at which your support team operates. Escalation rules only run during the business hours with which they are associated. To set business hours to 24-hours a day, set the times from midnight to midnight ($00:00:00 \sim 00:00:00$) on each day.

By default, business hours are set from 12:00 AM to 12:00 AM in the default time zone specified in your organization's profile.

BusinessProcess

Represents a business process.

Corresponds to an SObject4 in which the ObjectType="BusinessProcess".

Supported Calls

Create(),Update(),Query(),Retrieve(),GetUpdated(),CreateObject(),DescribeSObjects()

Special Access Rules

Customer Portal users cannot access this object.

Field	Field Type	Field Properties	Description
Description	string	Create	Description of this business process. Limit: 255 characters.
		Filter	

Field	Field Type	Field Properties	Description
		Nillable	
		Update	
IsActive boolean	boolean	Defaulted on create	Indicates whether this business process can be presented to users in the Salesforce user interface (true) or not (false)
		Filter	when creating a new record type or changing the business process of an existing record type.
		Update	
Name	string	Create	Required. Name of this business process. Limit: 80 characters.
		Filter	
		Update	
TableEnumOrId	picklist	Create	Required. One of the following values: Case, Opportunity,
		Filter	or Solution. Label is Entity Enumeration Or ID.
		Restricted picklist	

Use the BusinessProcess object to offer different subsets of picklist values to different users for the LeadStatus, CaseStatus, and OpportunityStage fields. Similar to a RecordType, a BusinessProcess identifies the type of a row in a Case, Lead, or Opportunity and implies a subset of picklist values for these three fields. The values for the remaining picklist fields are driven off of RecordType.

CallCenter

This object represents a call center, which is a logical representation of a single computer telephony integration (CTI) system instance in an organization.

Supported Calls

```
Create(),Query(),Retrieve(),GetDeleted(),GetUpdated(),DescribeSObject(),
DescribeSObjects(),GetUpdated()
```

Special Access Rules

Customer Portal users cannot access this object.

Field	Field Type	Field Properties	Description
CreatedById	reference	Defaulted on create	System field that contains the ID of the user who created this call center.
		Filter	
CreatedDate	dateTime	Defaulted on create	System field that contains the date and time that this call center was created.
		Filter	
Id	ID	Defaulted on create	System field that uniquely identifies this call center. Label is Call Center ID . This ID is created automatically when
		Filter the call	the call center is created.
InternalName	string	Create	The internal name of the call center.
		Filter	Limit is 255 characters.
		Iniliable	
LastModifiedById	reference	Defaulted on create	System field that contains the ID of the user who last modified this call center.
		Filter	
LastModifiedDate	dateTime	Defaulted on create	System field that contains the date and time this call center was last modified.
		Filter	
Name	string	Create	The name of the call center.
		Filter	Limit is 80 characters.
SystemModstamp	dateTime	Defaulted on create	System field that contains the date and time that this call center was last modified by a user or a workflow process
		Filter	conter that modified by a door of a worknow process.

Usage

Create a call center or query an existing call center.

Campaign

Represents and tracks a marketing campaign, such as a direct mail promotion, webinar, or trade show.

Corresponds to an SObject4 in which the ObjectType="Campaign".

Supported Calls

```
Create(),Update(),Query(),Search(),Retrieve(),Undelete(),CreateObject(),GetDeleted(),
GetUpdated(),DescribeSObjects()
```

Special Access Rules

Customer Portal users cannot access this object.

Fields

The Campaign statistics fields are read-only, as in the Salesforce user interface. You cannot update the statistics via the API.

Field	Field Type	Field Properties	Description
ActualCost	currency	Create	Amount of money spent to run the campaign.
		Filter	
		Nillable	
		Update	
AmountAllOpportunities	currency	Filter	Amount of money in all opportunities associated with the campaign, including closed/won opportunities. Label is Total Value Opportunities .
AmountWonOpportunities	currency	Filter	Amount of money in closed or won opportunities associated with the campaign. Label is Total Value Won Opportunities .
BudgetedCost	currency	Create	Amount of money budgeted for the campaign.
		Filter	
		Nillable	
		Update	
Description	textarea	Create	Description of the campaign. Limit: 32 KB. Only
		Nillable	the first 255 characters display in reports.
		Update	
EndDate	date	Create	Ending date for the campaign. Responses received
		Filter	after this date are still counted.
		Nillable	
		Update	
ExpectedResponse	percent	Create	Percentage of responses you expect to receive for
		Filter	the campaign.
		Nillable	
		Update	

Field	Field Type	Field Properties	Description
ExpectedRevenue	currency	Create Filter Nillable Update	Amount of money you expect to generate from the campaign.
HierarchyActualCost	currency	Filter	Calculated field for the total amount of money spent to run the campaigns in a campaign hierarchy. Label is Total Actual Cost in Hierarchy .
HierarchyBudgetedCost	currency	Filter	Calculated field for the total amount of money budgeted for the campaigns in a campaign hierarchy. Label is Total Budgeted Cost in Hierarchy .
HierarchyExpectedRevenue	currency	Filter	Calculated field for the total amount of money you expect to generate from the campaigns in a campaign hierarchy. Label is Total Expected Revenue in Hierarchy .
HierarchyNumberSent	int	Filter	Calculated field for the total number of individuals targeted by the campaigns in a campaign hierarchy. For example, the number of email messagess sent. Label is Total Num Sent in Hierarchy .
IsActive	boolean	Create Filter Nillable Update	Indicates whether this campaign is active (true) or not (false). Default value is false. Label is Active.
IsDeleted	boolean	Defaulted on create Filter	Indicates whether the object has been moved to the Recycle Bin (true) or not (false). Label is Deleted .
LastActivityDate	date	Filter Nillable	 Value is one of the following, whichever is the most recent: Due date of the most recent event logged against the record. Due date of the most recently closed task associated with the record.
Name	string	Create Filter Update	Required. Name of the campaign. Limit: is 80 characters.
NumberOfContacts	int	Filter	Number of contacts associated with the campaign. Label is Total Contacts .

Field	Field Type	Field Properties	Description
NumberOfConvertedLeads	int	Filter	Number of leads that were converted to an account and contact due to the marketing efforts in the campaign. Label is Converted Leads .
NumberOfLeads	int	Filter	Number of leads associated with the campaign. Label is Total Leads .
NumberOfOpportunities	int	Filter	Number of opportunities associated with the campaign. Label is Num Total Opportunities .
NumberOfResponses	int	Filter	Number of contacts and unconverted leads with a Member Status equivalent to "Responded" for the campaign. Label is Total Responses .
NumberOfWonOpportunities	int	Filter	Number of closed or won opportunities associated with the campaign. Label is Num Won Opportunities .
NumberSent	double	ible Create	Number of individuals targeted by the campaign.
		Filter	For example, the number of emails sent. Label is Num Sent.
		Nillable	Tum bent.
		Update	
OwnerId	reference	Create	ID of the user who owns this campaign. Default
		Defaulted on create	value is the user logging in to the API to perform the create.
		Filter	
		Update	
ParentCampaign	reference	Create	The campaign above the selected campaign in the
		Filter	campaign hierarchy.
		Nillable	
		Update	
StartDate	date	Create	Starting date for the campaign.
		Filter	
		Nillable	
		Update	
Status	picklist	Create	Status of the campaign, for example, Planned, In
		Filter	Progress. Limit: 40 characters.
		Nillable	
		Update	
TotalAmountAllOpportunities	currency	Filter	Calculated field for total amount of all opportunities associated with the campaign

Field	Field Type	Field Properties	Description
			hierarchy, including closed/won opportunities. Label is Total Value Opportunities in Hierarchy .
TotalAmountAllWonOpportunities	currency	Filter	Calculated field for amount of all closed/won opportunities associated with the campaign hierarchy. Label is Total Value Won Opportunities in Hierarchy .
TotalNumberofContacts	int	Filter	Calculated field for number of contacts associated with the campaign hierarchy. Label is Total Contacts in Hierarchy .
TotalNumberofConvertedLeads	int	Filter	Calculated field for the total number of leads associated with the campaign hierarchy that were converted into accounts, contacts, and opportunities. Label is Total Converted Leads in Hierarchy .
TotalNumberofLeads	int	Filter	Calculated field for total number of leads associated with the campaign hierarchy. This number also includes converted leads. Label is Total Leads in Hierarchy .
TotalNumberofOpportunties	int	Filter	Calculated field for the total number of opportunities associated with the campaign hierarchy. Label is Total Opportunities in Hierarchy .
TotalNumberofResponses	int	Filter	Calculated field for number of contacts and unconverted leads that have a Member Status equivalent to "Responded" for the campaign hierarchy. Label is Total Responses in Hierarchy .
TotalNumberofWonOpportunities	int	Filter	Calculated field for the total number of won opportunities associated with the campaign hierarchy. Label is Total Won Opportunities in Hierarchy
Туре	picklist	Create	Type of campaign, for example, Direct Mail or
		Filter	Referral Program. Limit: 40 characters.
		Nillable	
		Update	

Client applications can Create(), Update(), and Query() Attachments associated with a campaign via the API.

The Campaign object is defined only for those organizations that have the marketing feature enabled and valid marketing licenses. In addition, it is accessible only to those users that are enabled as marketing users. If the organization does not have the marketing feature or valid marketing licenses, this object is not available.DescribeGlobal() call, and you cannot use DescribeSObjects() or Query() with the Campaign object



Note: The main constituent of a campaign is a CampaignMember. You will commonly need to update campaigns with CampaignMember.

CampaignMember

Represents the association between a Campaign and either a Lead or Contact.

Corresponds to an SObject4 in which the ObjectType="CampaignMember".

Supported Calls

Create(),Update(),DescribeSObjects(),Query(),Retrieve(),CreateObject(),GetDeleted(), GetUpdated()

Special Access Rules

Customer Portal users cannot access this object.

Field	Field Type	Field Properties	Description
CampaignId	reference	Create	Required. ID of the Campaign to which this Lead or
		Filter	Contact is associated. For information on IDs, see ID Field Type.
ContactId	reference	Create	Required. ID of the Contact who is associated with a
		Filter	Campaign.
		Nillable	
FirstRespondedDate	date	Filter	Indicates the date on which the campaign member was first
		Nillable	given a responded status.
HasResponded	boolean	Defaulted on create	Indicates whether the campaign member has responded to the campaign (true) or not (false). Label is Responded .
		Filter	
IsDeleted	boolean	Defaulted on create	Indicates whether the object has been moved to the Recycle Bin (true) or not (false). Label is Deleted .
		Filter	
LeadId	reference	Create	Required. ID of the Lead who is associated with a Campaign.
		Filter	
		Nillable	
Status	picklist	Create	Controls the HasResponded flag on this object. You cannot
		Filter	directly set the HasResponded flag, as it is read-only, bu

Field	Field Type	Field Properties	Description
		Nillable Update	you can set it indirectly by setting this field in a create or update call. Each predefined value implies a HasResponded flag value. Each time you update this field, you implicitly update the HasResponded flag. In the Salesforce user interface, Marketing Users can define valid status values for the Status picklist. They can choose one status as the default status. For each Status field value, they can also select which values should be counted as "Responded," meaning that the HasResponded flag will be set to true for those values. 40 character limit.



Note: If you are importing CampaignMember data into Salesforce and need to set the value for an audit field, such as CreatedDate, contact salesforce.com. Audit fields are automatically updated during API operations unless you request to set these fields yourself. For more information, see System Fields.

Usage

Each record has a unique ID, and must contain either a ContactId or a LeadId, but cannot contain both. Any attempt to create a single record with both results in a successful insert but only the ContactId will be inserted. However, you can create two separate records on a Campaign—one for the Lead and one for the Contact.

This object is defined only for those organizations that have the marketing feature and valid marketing licenses. In addition, the object is accessible only to those users that are enabled as marketing users. If the organization does not have the marketing feature or valid marketing licenses, this object is not availabledoes not appear in the DescribeGlobal() call, and you cannot use DescribeSObjects() or Query() with the CampaignMember object.

You can indirectly update records by sending a Create() request. A Create() call is interpreted as an auto-insert-or-update call. The API determines whether a record exists with the specified CampaignId and ContactId or LeadId. If the record does not exist for the given ContactId or LeadId, then a Create() is performed. If the record already exists, the call is interpreted as an Update() and the Status field and HasResponded field on the existing record are updated. Thus, you cannot create duplicate records, because any attempt to create a duplicate record simply updates the existing record.

In API versions 15.0 and earlier, if you submit multiple records using a single Create () call, and if more than one record matches an existing record, only the first record submitted updates the existing record. If any of the submitted records match each other but do not match existing records, only the last record submitted is created.

During a Create() or Update() call, the Status field value specified in the call is verified as a valid status for the given Campaign:

- If the specified Status value is a valid status, the value is updated, and the HasResponded field is updated to either true or false, depending on the Status value association with HasResponded.
- If the specified Status value is not a valid status, the API assigns the default status to the Status field and updates the HasResponded field with the associated value. However, if the given Campaign does not have a default status, the API assigns the value specified in the call to the Status field, and the HasResponded field is set to false.

CampaignMemberStatus

One or more member status values defined for a campaign.

Supported Calls

```
Create(),Query(),Retrieve(),Update()
```

Special Access Rules

Customer Portal users cannot access this object.

You cannot delete a CampaignMemberStatus if that status is designated as the default status or if the status is currently used in a Campaign.

Fields

Field	Field Type	Field Properties	Description
CampaignId	reference	Create	ID of the campaign associated with this member status.
		Filter	
HasResponded	boolean	Create	Indicates whether this status is equivalent to "Responded"
		Defaulted on create	(true) or not (false).
		Filter	
		Update	
IsDefault	boolean	Create	Indicates whether this status is the default status (true) or
		Defaulted on create	not(false).
		Filter	
		Update	
IsDeleted	boolean	Defaulted	Indicates whether the object has been moved to the Recycle $\operatorname{Rig}(t, r_{\mathrm{NVO}})$ or pat (folloc). I shall in Dalated
		Filtor	bin (true) of not (rarse). Laber is Deleted.
		riiter	
Label	string	Create	Label for the status in the picklist. Limited to 765 characters.
		Filter	
SortOrder	int	Create	Order where this campaign member status appears in the
		Filter	picklist.
		Update	

Usage

Use this object to create picklist items for the member status in a campaign.

This object is defined only for those organizations that have the marketing feature and valid marketing licenses. In addition, the object is accessible only to those users that are enabled as marketing users. If the organization does not have the marketing feature or valid marketing licenses, this object is not available DescribeGlobal() call, and you cannot use DescribeSObjects() or Query() with the CampaignMember object.

CampaignOwnerSharingRule

Represents the rules for sharing a campaign with Users other than the owner or anyone above the owner in the role hierarchy. For more information, see "Setting Sharing Rules" in the Salesforce online help.



Note: Contact salesforce.com customer support to enable access to this object for your organization.

Supported Calls

Create(),Update(),DescribeSObjects(),Query(),Retrieve(),CreateObject(),GetDeleted(), GetUpdated()

Fields

Field	Field Type	Field Properties	Description
CampaignAccessLevel	picklist	Create Filter Restricted picklist Update	 A value that represents the type of access granted to the target Group, or UserRole. The possible values are: Read Edit All
GroupId	reference	Create Filter	The ID representing the source group. Campaigns owned by Users in the source Group trigger the rule to give access.
UserorGroupId	reference	Create Filter	The ID representing the User or Group being granted access.

Usage

Use this object to manage the sharing rules for a particular object.

CampaignShare

Represents a sharing entry on a Campaign.

Corresponds to an SObject4 in which the ObjectType="CampaignShare".

Supported Calls

Create(),Update(),DescribeSObjects(),Query(),Retrieve(),CreateObject()

Field	Field Type	Field Properties	Description
CampaignId	reference	Create Filter	ID of the Campaign associated with this sharing entry. This field cannot be updated. For information on IDs, see ID Field Type.
CampaignAccessLevel	picklist	Create Defaulted on create Filter Restricted picklist Update	 Level of access that the User or Group has to the Campaign. The possible values are: Read Edit All (This value is not valid for Create () or Update () calls.) This field must be set to an access level that is higher than the organization's default access level for Campaign.
IsDeleted	boolean	Defaulted on create Filter	Indicates whether the object has been moved to the Recycle Bin (true) or not (false). Label is Deleted .
RowCause	picklist	Filter Restricted picklist	 Reason that this sharing entry exists. Read-only. There are many possible values, including: Campaign Sharing Rule - The User or Group has access via a Campaign sharing rule. Manual Sharing - The User or Group has access because a User with "All" access manually shared the Campaign with them. Owner - The User is the owner of the Campaign or is in a Role above the Campaign owner in the role hierarchy.
UserOrGroupId	reference	Create Filter	ID of the User or Group that has been given access to the Campaign. This field cannot be updated.

Usage

This object allows you to determine which users and groups can view and/or edit Campaign records owned by other users.

CampaignTag

Associates a word or short phrase with a Campaign.

Supported Calls

Create(),Query(),Retrieve(),DescribeSObjects()

Field	Field Type	Field Properties	Description
ItemId	reference	Create Filter	ID of the tagged item.
Name	string	Create Filter	Name of the tag. If this value does not already exist, a new TagDefinition is created and becomes the parent of this Tag object. Otherwise, a TagDefinition with the same name becomes the parent of this Tag object. Parent relationships are created automatically.
TagDefinitionId	reference	Filter	ID of the parent TagDefinition object that owns the tag.
Туре	picklist	Create Filter Restricted picklist	 Defines the visibility of a tag. Possible value are: Public: The tag can be viewed and manipulated by all users in an organization Personal: The tag can be viewed or manipulated only by a user with a matching OwnerId

Usage

CampaignTag stores the relationship between its parent TagDefinition and the Campaign being tagged. Tag objects act as metadata, allowing users to describe and organize their data.

When a tag is deleted, its parent TagDefinition will also be deleted if the name is not being used; otherwise, the parent remains. Deleting a TagDefinition sends it to the recycle bin, along with any associated tag entries.

For more information on tags, see "About Tagging" in the Salesforce online help.

Case

Represents a case, which is a customer issue such as a customer's feedback, problem, or question.

Corresponds to an SObject4 in which the ObjectType="Case".

Supported Calls

```
Create(),Update(),DescribeSObjects(),Query(),Search(),Retrieve(),CreateObject(),
GetDeleted(),GetUpdated()
```

Field	Field Type	Field Properties	Description
AccountId	reference	Create Filter Nillable	ID of the account associated with this Case. For information about specifying this value, see Separating Accounts from Contacts in Cases.

Field	Field Type	Field Properties	Description
		Update	
CaseNumber	string	Autonumber	Assigned automatically when each case is inserted. It cannot
		Defaulted on create	be set directly, and it cannot be modified after the case is created.
		Filter	
ClosedDate	dateTime	Filter	The date and time when the case was closed.
		Nillable	
ContactId	reference	Create	ID of the associated Contact. For information on IDs, see
		Filter	ID Field Type.
		Nillable	
		Update	
Description	textarea	Create	A text description of the case. Limit: 32 KB.
		Nillable	
		Update	
HasCommentsUnreadByOwner	boolean	Defaulted on create	Indicates whether a case has comments that have not yet been read by the owner (true) or not (false).
		Filter	
HasSelfServiceComments	boolean	Defaulted on create	Indicates whether a case has comments added by a Self-Service user (true) or not (false).
		Filter	
IsClosed	boolean	Defaulted	Indicates whether the case is closed (true) or open (false).
		on create Filter	directly. Label is Closed .
IsClosedOnCreate	boolean	Defaulted	Indicates whether the case was closed at the same time that
		Filter	and is automatically set upon Create (). It cannot be set
		Filter	to true unless the IsClosed flag is also true.
IsDeleted	boolean	Defaulted on create	Indicates whether the object has been moved to the Recycle Bin (true) or not (false). Label is Deleted .
		Filter	
IsEscalated	boolean	Create	Indicates whether the case has been escalated (true) or
		Defaulted	A case's escalated state does not affect how you can use a $a = a = a = a = a = a = a = a = a = $
		on create Filter	However, you cannot set this flag via the API. For more information on case escalation, see the Salesforce online help. Label is Escalated .

Field	Field Type	Field Properties	Description
		Update	
IsSelfServiceClosed	boolean	Defaulted on create	Indicates whether the case is closed for Self-Service users (true) or not (false).
		Filter	
IsVisibleInSelfService	boolean	Defaulted on create	Indicates whether the case can be viewed in the Customer Self-Service Portal (true) or not (false).
		Filter	
Origin	picklist	Create	The source of the case, such as "Email," "Phone," or "Web."
		Filter	Label 15 Case Origin .
		Nillable	
		Update	
OwnerId	reference	Create	ID of the contact who owns the case.
		Defaulted on create	
		Filter	
		Update	
ParentId	reference	Create	The ID of the parent case in the hierarchy. The label is
		Filter	Parent Case.
		Nillable	
		Update	
Priority	picklist	Create	The importance or urgency of the case, such as "High,"
		Filter	"Medium," or "Low."
		Nillable	
		Update	
Reason	picklist	Create	The reason why the case was created, such as "Instructions
		Filter	not clear," or "User didn't attend training."
		Nillable	
		Update	
Status	picklist	Create	The status of the case, such as "New," "Closed," or
		Defaulted on create	"Escalated." This field directly controls the IsClosed flag. Each predefined Status value implies an IsClosed flag value For more information see CaseStatus
		Filter	autor for more mormation, see CaseOtatus.
		Nillable	
		Update	

Field	Field Type	Field Properties	Description
Subject	string	Create	The subject of the case. Limit: 255 characters.
		Filter	
		Nillable	
		Update	
SuppliedCompany	string	Create	The company name that was entered when the case was
		Filter	created. Cannot be updated after the case has been created Label is Company .
		Nillable	1 7
		Update	
SuppliedEmail	email	Create	The email address that was entered when the case was created.
	Create	Filter	Cannot be updated after the case has been created. Label is Email .
	Filter	Nillable	If your organization has an active auto-response rule,
	Nillable	Update	SuppliedEmail is required when creating a case via the
Update API. Auto-response specified by Contac record, the email spe see "Setting Up Au online help.	API. Auto-response rules use the email in the contact specified by ContactId. If no email address is in the contact record, the email specified here is used. For more information, see "Setting Up Auto-Response Rules" in the Salesforce online help.		
SuppliedName	string	Create	The name that was entered when the case was created.
		Filter	Cannot be updated after the case has been created. Label is Name .
		Nillable	
		Update	
SuppliedPhone	string	Create	The phone number that was entered when the case was
		Filter	created. Cannot be updated after the case has been created. Label is Phone .
		Nillable	
		Update	
Туре	picklist	Create	The type of case, such as "Feature Request" or "Question."
		Filter	
		Nillable	
		Update	



Note: If you are importing Case data into Salesforce and need to set the value for an audit field, such as CreatedDate, contact salesforce.com. Audit fields are automatically updated during API operations unless you request to set these fields yourself. For more information, see System Fields.

Use the Case object to manage cases for your organization. Client applications can Query() or Update(), and Query() Attachments associated with a case via the API.

Assignment Rules

When you Create() or Update() a case, your client application can have the case automatically assigned to one or more Users based on assignment rules that have been configured in the Salesforce user interface. To use this feature, your client application needs to set either of the following options (but not both) in the SOAP header (see SetSOAPHeader()) used in the Create() or Update() call:

Field	Field Type	Description
assignmentRuleId	reference	ID of the assignment rule to use. Can be an inactive assignment rule. If unspecified and useDefaultRule is true, then the default assignment rule is used. To find the ID for a given assignment rule, query the AssignmentRule object (specifying RuleType="caseAssignment"), iterate through the returned AssignmentRule objects, find the one you want to use, retrieve its ID, and then specify its ID in this field in the SetSOAPHeader().
useDefaultRule	boolean	Specifies whether to use the default rule for rule-based assignment (true) or not (false). The default rule is assigned by users in the Salesforce user interface.

Separating Accounts from Contacts in Cases

In releases before 8.0, the AccountId could not be specified, it was derived from the contact's account. This behavior will continue to be supported in future releases, but you can also now specify an AccountId. If you do not specify the AccountId during the creation of a case, the value will default to the contact's AccountId.



Note: When a record is updated, if the ContactId has not changed, then the AccountId is not regenerated. This prevents the API from overwriting a value previously changed in the Salesforce user interface. However, if an API call changes the ContactId and the AccountId field is empty, then the AccountId is generated using the contact's account.

Using _case with Java

Depending on the development tool you use, you may need to write your application using _case instead of Case. This is because case is a reserved word in Java.

CaseComment

Represents a comment that provides additional information about the associated Case.

Corresponds to an SObject4 in which the ObjectType="CaseComment".

```
Create(),Update(),Query(),Search(),Retrieve(),CreateObject(),GetDeleted(),GetUpdated(),
DescribeSObjects()
```

Field	Field Type	Field Properties	Description
CommentBody	textarea	Create Filter Nillable	Text of the CaseComment. The maximum size of the comment body is 4000 bytes. Label is Body .
IsDeleted	boolean	Defaulted on create Filter	Indicates whether the object has been moved to the Recycle Bin (true) or not (false). Label is Deleted .
IsPublished	boolean	Create Filter Defaulted on create Update	Indicates whether the CaseComment is visible to customers in the Self-Service portal (true) or not (false). Label is Published . This is the only CaseComment field that can be updated via the API.
ParentId	reference	Create Filter	Required. ID of the parent Case of the CaseComment. For information on IDs, see ID Field Type.



Note: If you are importing CaseComment data into Salesforce and need to set the value for an audit field, such as CreatedDate, contact salesforce.com. Audit fields are automatically updated during API operations unless you request to set these fields yourself. For more information, see System Fields.

Usage

In the Salesforce user interface, comments are generally entered by Users working on a Case. All users have access to create and view CaseComments in the Salesforce user interface and when using the API. In the API, CaseComments cannot be modified after insertion unless the user has the "Modify All" object-level permission for Cases or the "Modify All Data" permission. If not, users can only update the IsPublished field, and cannot delete CaseComments.

CaseContactRole

Represents the role that a given Contact plays on a Case.

Corresponds to an SObject4 in which the ObjectType="CaseContactRole".

```
Create(),Update(),DescribeSObjects(),Query(),Retrieve(),CreateObject(),GetDeleted(),
GetUpdated()
```

Field	Field Type	Field Properties	Description
CaseId	reference	Create Filter	ID of the Case associated with this Contact. For information on IDs, see ID Field Type.
ContactId	reference	Create Filter Update	Required. ID of the Contact. For information on IDs, see ID Field Type.
IsDeleted	boolean	Defaulted on create Filter	Indicates whether the object has been moved to the Recycle Bin (true) or not (false). Label is Deleted .
Role	picklist	Create Filter Nillable Update	Name of the role played by the Contact on this Contract, such as Decision Maker, Approver, Buyer, and so on. Must be unique—there cannot be multiple records in which the ContractId, ContactId, and Role values are identical. Different contacts can play the same role on the same contract. A contact can play different roles on the same contract.

Usage

Use this object to define the role that a given Case plays on a given Contact. For example, you can use this object to be able to see all contacts who are associated to a case, or, given a contact, be able to query all cases that they are associated with, even if they are not the primary contact on the case.

CaseHistory

Represents historical information about changes that have been made to the associated Case.

Corresponds to an SObject4 in which the ObjectType="CaseHistory".

Supported Calls

Query(),Retrieve(),GetDeleted(),GetUpdated(),CreateObject(),DescribeSObjects()

Special Access Rules

This object is always read-only in Salesforce.

Field	Field Type	Field Properties	Description
CaseId	reference	Filter	ID of the Case associated with this record. For information on IDs, see ID Field Type.
Field	picklist	Filter Restricted picklist	 Name of the case field that was modified, or a special value to indicate some other modification to the case. The possible values, in addition to the case field names, are: ownerAssignment - The owner of the case was changed. ownerAccepted - A user took ownership of a case from a queue. ownerEscalated - The owner of the case was changed due to case escalation. external - A user made the case visible to customers in the Customer Self-Service Portal.
IsDeleted	boolean	Defaulted on create Filter	Indicates whether the object has been moved to the Recycle Bin (true) or not (false). Label is Deleted .

Usage

Case history entries are indirectly created by modifying a case via the Salesforce user interface or the API.

Two rows are added to this record when foreign key fields change. One row contains the foreign key object names that display in the online application. For example, "Jane Doe" is recorded as the name of a Contact. The other row contains the actual foreign key ID that is only returned to and visible from the API.

This object respects field level security on the parent object.

CaseOwnerSharingRule

Represents the rules for sharing a case with users other than the owner.



Note: Contact salesforce.com customer support to enable access to this object for your organization.

Supported Calls

```
Create(),Update(),DescribeSObjects(),Query(),Retrieve(),CreateObject(),GetDeleted(),
GetUpdated()
```

Special Access Rules

Customer Portal users cannot access this object.

Field	Field Type	Field Properties	Description
CaseAccessLevel	picklist	Create Filter Restricted picklist Update	 A value that represents the type of sharing being allowed. The possible values are: Read Edit
GroupId	reference	Create Filter	The ID representing the source group. Cases owned by users in the source group trigger the rule to give access.
UserorGroupId	reference	Create Filter	The ID representing the target user or group. Target users or groups are given access.

Usage

Use this object to manage the sharing rules for a particular case. General sharing and territory management-related sharing use this object.

CaseShare

Represents a sharing entry on a Case.

Corresponds to an SObject4 in which the ObjectType="CaseShare".

Supported Calls

Create(),Update(),DescribeSObjects(),Query(),Retrieve(),CreateObject()

Special Access Rules

Customer Portal users cannot access this object.

Field	Field Type	Field Properties	Description
CaseAccessLevel	picklist	Filter Restricted picklist	 Level of access that the User or Group has to the Case. The possible values are: Read Edit All (This value is not valid for Create() or Update() calls.)

Field	Field Type	Field Properties	Description
			This field must be set to an access level that is higher than the organization's default access level for cases.
CaseId	reference	Filter	ID of the Case associated with this sharing entry. This field cannot be updated. For information on IDs, see ID Field Type.
IsDeleted	boolean	Defaulted on create Filter	Indicates whether the object has been moved to the Recycle Bin (true) or not (false). Label is Deleted .
RowCause	picklist	Filter Restricted picklist	 Reason that this sharing entry exists. Read-only. Values may include: Manual - The User or Group has access because a user with "All" access manually shared the Case with them. Owner - The User is the owner of the Case or is in a Role above the Case owner in the role hierarchy. ImplicitChild - User or Group has access to the Case on the Account associated with this Case. Rule - The User or Group has access via a Case sharing rule.
UserOrGroupId	reference	Filter	ID of the User or Group that has been given access to the Case. This field cannot be updated.

This object allows you to determine which users and groups can view and edit Cases owned by other users. For more information, see Sharing.

If you attempt to create a new record that matches an existing record, the Create() call updates any modified fields and returns the existing record.

CaseSolution

Represents the association between a Case and a Solution.

Corresponds to an SObject4 in which the ObjectType="CaseSolution".

```
Create(),DescribeSObjects(),Query(),Retrieve(),CreateObject(),GetDeleted(),GetUpdated()
```

Field	Field Type	Field Properties	Description
CaseId	reference	Create Filter	Required. ID of the Case associated with the Solution. For information on IDs, see ID Field Type.
IsDeleted	boolean	Defaulted on create Filter	Indicates whether the object has been moved to the Recycle Bin (true) or not (false). Label is Deleted .
SolutionId	reference	Create Filter	Required. ID of the Solution associated with the case.

Usage

You cannot Update () this object via the API. If you attempt to create a record that matches an existing record, the Create () call simply returns the existing record.

CaseStatus

Represents the status of a Case, such as New, On Hold, or In Process.

Corresponds to an SObject4 in which the ObjectType="CaseStatus".

Supported Calls

Query(),Retrieve(),CreateObject(),DescribeSObjects()

Field	Field Type	Field Properties	Description
isClosed	boolean	Defaulted on create Filter	Indicates whether this case status value represents a closed Case (true) or not (false). Multiple case status values can represent a closed Case.
IsDefault	boolean	Defaulted on create Filter	Indicates whether this is the default case status value (true) or not (false) in the picklist.
MasterLabel	string	Filter Nillable	Master label for this case status value. This display value is the internal label that does not get translated.
SortOrder	int	Filter Nillable	Number used to sort this value in the case status picklist. These numbers are not guaranteed to be sequential, as some previous case status values might have been deleted.

This object represents a value in the case status picklist. The case status picklist provides additional information about the status of a Case, such as whether a given Status value represents an open or closed case. Your client application can invoke the Query () call on the CaseStatus object to retrieve the set of values in the case status picklist, and then use that information while processing Case objects to determine more information about a given case. For example, the application could test whether a given case is open or closed based on its Status value and the value of the isclosed property in the associated CaseStatus object.

The CaseStatus object is read-only via the API. With sufficient permissions, your client application can invoke the Query() and DescribeSObjects() calls on this object.

CaseTag

Associates a word or short phrase with a Case.

Supported Calls

Create(),Query(),Retrieve(),DescribeSObjects()

Fields

Field	Field Type	Field Properties	Description
ItemId	reference	Create Filter	ID of the tagged item.
Name	string	Create Filter	Name of the tag. If this value does not already exist, a new TagDefinition is created and becomes the parent of this Tag object. Otherwise, a TagDefinition with the same name becomes the parent of this Tag object. Parent relationships are created automatically.
TagDefinitionId	reference	Filter	ID of the parent TagDefinition object that owns the tag.
Туре	picklist	Create Filter Restricted picklist	 Defines the visibility of a tag. Possible value are: Public: The tag can be viewed and manipulated by all users in an organization Personal: The tag can be viewed or manipulated only by a user with a matching OwnerId

Usage

CaseTag stores the relationship between its parent TagDefinition and the Case being tagged. Tag objects act as metadata, allowing users to describe and organize their data.

When a tag is deleted, its parent TagDefinition will also be deleted if the name is not being used; otherwise, the parent remains. Deleting a TagDefinition sends it to the recycle bin, along with any associated tag entries.

For more information on tags, see "About Tagging" in the Salesforce online help.

CaseTeamMember

Represents a case team member, who works with a team of other users to help resolve a case.

Supported Calls

Create(),Update(),DescribeSObjects(),Query(),Retrieve(),GetDeleted(),GetUpdated()

Fields

Field	Field Type	Field Properties	Description
MemberId	reference	Create Filter	The ID of the user or contact who is a member on a case team.
ParentId	reference	Create Filter	The ID of the case with which the case team member is associated.
TeamRoleId	reference	Create Filter Update	The ID of the case team role with which the case team member is associated.
TeamTemplateMemberId	reference	Filter Nillable	The ID of the team member included in a predefined case team.

CaseTeamRole

Represents a case team role. Every case team member has a role on a case, such as "Customer Contact" or "Case Manager."

Supported Calls

Create(),Update(),Query(),Retrieve(),DescribeSObjects(),GetDeleted(),GetUpdated()

Field	Field Type	Field Properties	Description
AccessLevel	picklist	Create Filter Restricted picklist Update	 A value that represents the type of access granted to the target Group for cases. The possible values are: None Read Edit

Field	Field Type	Field Properties	Description
Name	string	Create	The name of the case team role.
		Filter	
		Update	
PreferencesVisibleinCSP	boolean	Create	Indicates whether or not the case team role is visible to
		Update	Customer Portal users.

CaseTeamTemplate

Represents a predefined case team, which is a group of users that helps resolve a case.

Supported Calls

Create(),Update(),DescribeSObjects(),Query(),Retrieve(),GetDeleted(),GetUpdated()

Fields

Field	Field Type	Field Properties	Description
Description	textarea	Create Filter Nillable Update	A text description of the predefined case team.
Name	string	Create Filter Update	The name of the predefined case team.

CaseTeamTemplateMember

Represents a member on a predefined case team, which is a group of users that helps resolve cases.

Supported Calls

Create(),Update(),DescribeSObjects(),Query(),Retrieve(),GetDeleted(),GetUpdated()

Field	Field Type	Field Properties	Description
MemberId	reference	Create Filter	The ID of the user or contact who is a team member on a predefined case team.
TeamRoleId	reference	Create Filter Update	The ID of the predefined case team member's case team role.
TeamTemplateMemberId	reference	Filter Nillable	The ID of the team member included in a predefined case team.

CaseTeamTemplateRecord

The CaseTeamTemplateRecord object is a linking object between the Case and CaseTeamTemplate objects. To assign a predefined case team to a case (customer inquiry), create a CaseTeamTemplateRecord object and point the ParentId to the case and the TeamTemplateId to the predefined case team.

Supported Calls

Create(),DescribeSObjects(),Query(),Retrieve(),GetDeleted(),GetUpdated()

Fields

Field	Field Type	Field Properties	Description
ParentId	reference	Create Filter	The ID of the case with which the case team template record is associated.
TeamTemplateId	reference	Create Filter	The ID of the predefined case team with which the case team template record is associated.

CategoryData

Represents a logical grouping of Solution records.

Corresponds to an SObject4 in which the ObjectType="CategoryData".

Supported Calls

```
Create(),Update(),Query(),Retrieve(),CreateObject(),DescribeSObjects(),GetDeleted(),
GetUpdated()
```

Special Access Rules

Customer Portal users cannot access this object.

Fields

Field	Field Type	Field Properties	Description
CategoryNodeId	reference	Create Filter Update	ID of the CategoryNode associated with the solution.
IsDeleted	boolean	Defaulted on create Filter	Indicates whether the object has been moved to the Recycle Bin (true) or not (false). Label is Deleted .
RelatedSobjectId	reference	Create Filter Update	ID of the solution related to the category.

Usage

This object allows you to assign one or more categories to a Solution. It is an intermediate data table with two foreign keys that defines the relationship between a CategoryNode and a Solution record.

CategoryData has two foreign keys:

- The first foreign key, CategoryNodeId, refers to the ID of a CategoryNode.
- The other foreign key, RelatedSobjectId, refers to a Solution ID.

This is a many-to-many relationship, so there can be multiple rows returned with a CategoryNodeId. A Solution can be associated with multiple categories.

CategoryNode

Represents a tree of Solution categories.

Corresponds to an SObject4 in which the ObjectType="CategoryNode".

```
Create(),Update(),DescribeSObjects(),Query(),Retrieve(),CreateObject(),GetDeleted(),
GetUpdated()
```

Special Access Rules

- Customer Portal users cannot access this object.
- Attempting to delete a CategoryNode that has children (referred by CategoryNode.Parent), or is referred to elsewhere, will cause a failure.

Fields

Field	Field Type	Field Properties	Description
MasterLabel	string	Create	Label for the category node.
		Filter	
		Update	
ParentId	reference	Create	ID of the parent of this node, if any.
		Filter	
		Nillable	
		Update	
SortOrder	int	Create	Indicates the sort order of child CategoryNode objects.
		Filter	
		Nillable	
		Update	
SortStyle	picklist	Create	Indicates whether the sort order is alphabetical or custom.
		Filter	
		Restricted picklist	
		Update	

Usage

A CategoryNode defines a category of solutions. In the Salesforce user interface, you can edit category definitions at Setup ➤ Customize ➤ Solution ➤ Solution Categories.

CategoryNodeLocalization

When the translation workbench is enabled for your organization, the CategoryNodeLocalization object provides the translation of the label of a category. For information on the translation workbench, see the Salesforce online help.

```
Create(),Update(),Query(),Retrieve(),CreateObject(),DescribeSObjects(),GetDeleted(),
GetUpdated()
```

Special Access Rules

- Your organization must be using Professional, Enterprise, Developer, or Unlimited Edition and be enabled for the translation workbench.
- To view this object, you must have the "View Setup and Configuration" permission.

Field	Field Type	Field Properties	Description
CategoryNodeId	reference	Create Filter Nillable Restricted picklist	The combined language and locale ISO code, which controls the language for labels displayed in an application. This picklist contains the following labels and values: • English: en_US • German: de • Spanish: es • French: fr • Italian: it • Japanese: ja • Swedish: sv • Korean: ko • Chinese (Traditional): zh_TW • Chinese (Traditional): zh_TW • Chinese (Simplified): zh_CN • Portuguese (Brazilian) pt_BR • Dutch: nl_NL • Danish: da • Thai: th • Finnish: fi • Russian: ru The following languages are also available if you request them from your salesforce.com representative: • Czech: cs • Hungarian: hu • Indonesian: in • Polish: pl • Turkish: tr The values in this field are not related to the default locale selection.
CategoryNodeId	reference	Create Filter Nillable	The ID of the CategoryNode that is being translated.
Value	string	Create Filter Nillable	The actual translated label for the category. Label is Translation .

Field	Field Type	Field Properties	Description
		Update	

Use this object to translate the labels of your categories into the different languages supported by Salesforce. Users with the translation workbench enabled can view category node translations, but either the "Customize Application," "Manage Translation," or "Manage Categories" permission is required to Create() or Update() category node translations.

Community

Represents an Idea community.

Supported Calls

Query(),Retrieve()

Fields

Field	Field Type	Field Properties	Description
Description	string	Create	Text description of the Community.
		Filter	
		Nillable	
		Update	
IsActive	boolean	Create	Indicates whether the Community is active or inactive. A
		Filter	Idea can only be posted to an active Community.
		Update	
Name	string	Create	The name of the Community.
		Filter	
		Update	

Usage

Use this object to create communities that help organize Ideas into logical groups. For more information on ideas, see "About Ideas" in the Salesforce online help.

The boolean field IsPublished is not exposed in the public API for both entities. The value is defaulted at insert time based on whether the context user is a portal user or not. Soql queries also filter on this flag, so portal users can only see portal ideas and comments, and standard/admin users can only see internal ideas and comments.

Contact

Represents a contact, which is an individual associated with an account.

Corresponds to an SObject4 in which the ObjectType="Contact".

Supported Calls

Create(),Update(),Query(),Search(),Retrieve(),CreateObject(),GetDeleted(),GetUpdated(), DescribeSObjects()

Special Access Rules

Customer Portal users can access only portal-enabled contacts.

Field	Field Type	Field Properties	Description
AccountId	reference	Create	ID of the account that is the parent of this contact.
		Filter	
		Nillable	
		Update	
AssistantName	string	Create	The name of the assistant.
		Filter	
		Nillable	
		Update	
AssistantPhone	phone	Create	The telephone number of the assistant.
		Filter	
		Nillable	
		Update	
Birthdate	date	Create	The birthdate of the contact.
		Filter	
		Nillable	
		Update	
CanAllowPortalSelfReg	boolean	Create	Indicates whether this contact can self-register for your
		Defaulted on create	organization's Customer Portal(true) or not (talse). For more information about the Customer Portal and self-registration, see "Enabling Customer Portal Login and
		Filter	Settings" in the Salesforce online help.
		Update	

Field	Field Type	Field Properties	Description
ConnectionReceivedID	reference	Filter Nillable	ID of the PartnerNetworkConnection that shared this record with your organization. This field is only available if you have enabled Salesforce to Salesforce.
ConnectionSentID	reference	Filter Nillable	ID of the PartnerNetworkConnection that you shared this record with. This field is only available if you have enabled Salesforce to Salesforce. Beginning with API version 15.0, the ConnectionSentID field is no longer supported. The ConnectionSentID field is still be visible, but the value is null. You can use the new PartnerNetworkRecordConnection object to forward records to connections.
Department	string	Create Filter Nillable Update	The department of the contact.
Description	textarea	Create Nillable Update	A description of the contact. Label is Contact Description . Limit: 32 KB.
Email	email	Create Filter Nillable Update	Email address for the contact.
EmailBouncedDate	dateTime	Create Filter Nillable Update	If bounce management isactivated and an email sent to the contact bounces, the date and time the bounce occurred.
EmailBouncedReason	string	Create Filter Nillable Update	If bounce management is activated and an email sent to the contact bounces, the reason the bounce occurred.
Fax	phone	Create Filter Nillable Update	Fax number for the contact. Label is Business Phone .
FirstName	string	Create Filter	First name of contact. Limited to 40 characters.
Field	Field Type	Field Properties	Description
--------------------	------------	------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------
		Nillable	
		Update	
HasOptedOutofEmail	boolean	Create	Indicates whether the contact would prefer not to receive
		Defaulted on create	email from salesforce.com (true) or not (false). Label is Email Opt Out.
		Filter	
		Update	
HomePhone	phone	Create	Home telephone number for the contact.
		Filter	
		Nillable	
		Update	
IsDeleted	boolean	Defaulted on create	Indicates whether the object has been moved to the Recycle Bin (true) or not (false). Label is Deleted .
		Filter	
IsPersonAccount	boolean	Defaulted on create	Indicates whether this contact is the sole child contact of a person account (true) or not (false). For more information about person accounts, see Person Account
		Filter	Record Types and the Salesforce online help.
LastActivityDate	date	Filter	Value is one of the following, whichever is the most recent:
		Nillable	• Due date of the most recent event logged against the record.
			• Due date of the most recently closed task associated with the record.
LastCURequestDate	dateTime	Filter	The last date that a Stay-in-Touch request was sent to the
		Nillable	contact.
LastCUUpdateDate	dateTime	Filter	The last time a Stay-in-Touch update was processed for the
		Nillable	contact.
LastName	string	Create	Required. Last name of contact. Limited to 80 characters.
		Filter	
		Update	
LeadSource	picklist	Create	The source of the lead.
		Filter	
		Nillable	
		Update	

Field	Field Type	Field Properties	Description
 MailingCity MailingState MailingCountry MailingPostalCode 	string	Create Filter Nillable Update	Details for the mailing address.
MailingStreet	textarea	Create Filter Nillable Update	Street address for mailing address.
MasterRecordId	reference	Filter Nillable	If this object was deleted as the result of a merge, this field contains the ID of the record that was kept. If this object was deleted for any other reason, or has not been deleted, the value is null.
MobilePhone	phone	Create Filter Nillable Update	Contact's mobile phone number.
Name	string	Filter	Concatenation of FirstName and LastName. Limited to 121 characters.
 OtherCity OtherCountry OtherPostalCode OtherState 	string	Create Filter Nillable Update	Details for alternate address.
OtherPhone	phone	Create Filter Nillable Update	Telephone for alternate address.
OtherStreet	textarea	Create Filter Nillable Update	Street for alternate address.
OwnerId	reference	Create Defaulted on create Filter	The ID of the owner of the account associated with this contact.

Field	Field Type	Field Properties	Description
		Update	
Phone	phone	Create	Telephone number for the contact. Label is Business Phone .
		Filter	
		Nillable	
		Update	
ReportsToId	reference	Create	This field is not visible if IsPersonAccount is true.
		Filter	
		Nillable	
		Update	
Salutation	picklist	Create	Honorific abbreviation, word, or phrase to be used in front
		Filter	of name in greetings, such as Dr. or Mrs.
		Nillable	
		Update	
Title	string	Create	Title of the contact such as CEO or Vice President.
		Filter	
		Nillable	
		Update	



Note: If you are importing Contact data into Salesforce and need to set the value for an audit field, such as

CreatedDate, contact salesforce.com. Audit fields are automatically updated during API operations unless you request to set these fields yourself. For more information, see System Fields.

Usage

Use this object to manage individuals who are associated with an Account in your organization. Client applications can use the API to Create(), Query() or Update() any Attachment associated with a contact.

ContactHistory

Represents the history of changes to the values in the fields of a contact. This object is available in versions 11.0 and later.

Supported Calls

Query(),Retrieve(),GetDeleted(),GetUpdated(),DescribeSObjects()

Field	Field Type	Field Properties	Description
Field	picklist	Filter	The name of the field that was changed.
		Restricted picklist	
IsDeleted	boolean	Defaulted on create Filter	Indicates whether the object has been moved to the Recycle Bin (true) or not (false). Label is Deleted .
ContactId	reference	Filter	ID of the Contact. For information on IDs, see ID Field Type. Label is Contact ID.

Usage

Use this object to identify changes to a contact.

This object respects field level security on the parent object.

ContactOwnerSharingRule

Represents the rules for sharing a contact with Users other than the owner. For more information, see "Setting Sharing Rules" in the Salesforce online help.



Note: Contact salesforce.com customer support to enable access to this object for your organization.

Supported Calls

```
Create(),Update(),DescribeSObjects(),Query(),Retrieve(),CreateObject(),GetDeleted(),
GetUpdated()
```

Field	Field Type	Field Properties	Description
ContactAccessLevel	picklist	Create Filter Restricted picklist Update	A value that represents the type of access granted to the target Group, UserRole, or User for Contacts. The possible values are: • Read • Edit
GroupId	reference	Create Filter	The ID representing the source group. Contacts owned by Users in the source Group trigger the rule to give access.

Field	Field Type	Field Properties	Description
UserorGroupId	reference	Create	The ID representing the User or Group being granted access.
		Filter	

Use this object to manage the sharing rules for a particular object.

ContactShare

Represents a list of access levels to a Contact along with an explanation of the access level. For example, if you have access to a record because you own it, the Access Level is "All" and Reason for Access is "Owner."

Corresponds to an SObject4 in which the ObjectType="ContactShare".

Supported Calls

Create(),Update(),Query(),Retrieve(),CreateObject(),DescribeSObjects()

Special Access Rules

Customer Portal users cannot access this object.

Field	Field Type	Field Properties	Description
ContactId	reference	Create Filter	ID of the Contact associated with this sharing entry. This field cannot be updated. For information on IDs, see ID Field Type.
ContactAccessLevel	picklist	Create Defaulted on create Filter Restricted picklist Update	 Level of access that the User or Group has to cases associated with the account Contact. The possible values are: Read Edit All (This value is not valid for Create () or Update () calls.) This field must be set to an access level that is higher than the organization's default access level for contacts.
IsDeleted	boolean	Defaulted on create Filter	Indicates whether the object has been moved to the Recycle Bin (true) or not (false). Label is Deleted .

Field	Field Type	Field Properties	Description
RowCause	picklist	Filter Restricted picklist	 Reason that this sharing entry exists. Read-only. There are many possible values, including: Account Sharing - The User or Group has access because a User with "All" manually shared the Account associated with the contact. Account Sharing Rule - The User or Group has access via an Account sharing rule for the account associated with the contact. Account Team - The User has access to the Contact via his or her status as an AccountTeamMember. Contact Sharing Rule - The User or Group has access via a Contact sharing rule. ImplicitChild - The User or Group has access to the Contact via sharing access on the associated Account. Manual Sharing - The User or Group has access because a User with "All" access manually shared the Contact with them. Owner - The User is the owner of the Contact or is in a Role above the Contact owner in the role hierarchy.
UserOrGroupId	reference	Create Filter	ID of the User or Group that has been given access to the Contact. This field cannot be updated.

This object allows you to determine which users and groups can view and/or edit Contact records owned by other users.

ContactTag

Associates a word or short phrase with a Contact.

Supported Calls

```
Create(),Query(),Retrieve(),DescribeSObjects()
```

Field	Field Type	Field Properties	Description
ItemId	reference	Create Filter	ID of the tagged item.
Name	string	Create Filter	Name of the tag. If this value does not already exist, a new TagDefinition is created and becomes the parent of this Tag

Field	Field Type	Field Properties	Description
			object. Otherwise, a TagDefinition with the same name becomes the parent of this Tag object. Parent relationships are created automatically.
TagDefinitionId	reference	Filter	ID of the parent TagDefinition object that owns the tag.
Туре	picklist	Create Filter Restricted picklist	 Defines the visibility of a tag. Possible value are: Public: The tag can be viewed and manipulated by all users in an organization Personal: The tag can be viewed or manipulated only by a user with a matching OwnerId

Contact Tag stores the relationship between its parent TagDefinition and the Contact being tagged. Tag objects act as metadata, allowing users to describe and organize their data.

When a tag is deleted, its parent TagDefinition will also be deleted if the name is not being used; otherwise, the parent remains. Deleting a TagDefinition sends it to the recycle bin, along with any associated tag entries.

For more information on tags, see "About Tagging" in the Salesforce online help.

Contract

Represents a contract (a business agreement) associated with an Account.

Corresponds to an SObject4 in which the ObjectType="Contract".

Supported Calls

```
Create(),Update(),Query(),Search(),Retrieve(),CreateObject(),GetDeleted(),GetUpdated(),
DescribeSObjects(),
```

Field	Field Type	Field Properties	Description
AccountId	reference	Create	Required. ID of the Account associated with this contract.
		Filter	For information on IDs, see ID Field Type.
		Update	
ActivatedById	reference	Filter	ID of the User who activated this contract.
		Nillable	
ActivatedDate	dateTime	Filter	Date and time when this contract was activated.
		Nillable	

Field	Field Type	Field Properties	Description
 BillingCity BillingCountry BillingPostalCode BillingState 	string	Create Filter Nillable Update	Details for the billing address.
BillingStreet	textarea	Create Filter Nillable Update	Street address for the billing address.
CompanySignedDate	date	Create Filter Nillable Update	Date on which the contract was signed by your organization.
CompanySignedId	reference	Create Filter Nillable Update	ID of the User who signed the contract.
ContractNumber	string	Autonumber Defaulted on create Filter	Number of the contract.
ContractTerm	int	Create Filter Nillable Update	Number of months that the contract is valid.
CustomerSignedDate	date	Create Filter Nillable Update	Date on which the customer signed the contract.
CustomerSignedId	reference	Create Filter Nillable Update	ID of the Contact who signed this contract.

Field	Field Type	Field Properties	Description
CustomerSignedTitle	string	Create Filter	Title of the customer who signed the contract.
		Nillable	
		Update	
Description	textarea	Create	Description of the contract.
		Nillable	
		Update	
EndDate	date	Filter Nillable	Read-only. Calculated end date of the contract. This value is calculated by adding the ContractTerm to the StartDate.
IsDeleted	boolean	Defaulted on create	Indicates whether the object has been moved to the Recycle Bin (true) or not (false). Label is Deleted .
		Filler	
LastActivityDate	date	Filter Nillable	Value is one of the following, whichever is the most recent:Due date of the most recent event logged against the record.
			• Due date of the most recently closed task associated with the record.
LastApprovedDate	dateTime	Filter	Last date the contract was approved.
		Nillable	
OwnerExpirationNotice	picklist	Create	Number of days ahead of the contract end date $(15, 30, 45, 0)$
		Filter	60, 90, and 120). Used to notify the owner in advance that the contract is ending.
		Nillable	<u> </u>
		Restricted picklist	
		Update	
OwnerId	reference	Create	ID of the user who owns the contract.
		Defaulted on create	
		Filter	
		Nillable	
		Update	
Pricebook2Id	reference	Create	ID of the pricebook, if any, associated with this contract.
		Filter	
		Nillable	

Field	Field Type	Field Properties	Description
		Update	
 ShippingCity ShippingCountry ShippingPostalCode ShippingState 	string	Create Filter Nillable Update	Details for the shipping address.
ShippingStreet	textarea	Create Filter Nillable Update	Street address for the shipping address.
SpecialTerms	textarea	Create Filter Nillable Update	Special terms that apply to the contract.
StartDate	date	Create Filter Nillable Update	Start date for this contract. Label is Contract Start Date .
Status	picklist	Create Defaulted on create Filter Update	The picklist of values that indicate order status. Each value is within one of two status categories defined in StatusCode. For example, the status picklist may contain: Ready to Ship, Shipped, Received as values within the Activated StatusCode.
StatusCode	picklist	Filter Restricted picklist	The status category for the order. An order can be either Draft, InApproval, or Activated. Label is Status Category .

The Contract object represents a business agreement. Client applications can use the API to Create(), Update(), Query(), and Retrieve() contracts.

The Status field specifies the current state of a contract. Status strings (defined in the ContractStatus object) represent its current state (Draft, InApproval, or Activated).

Client applications must initially Create() a Contract in a non-Activated state. Client applications can subsequently activate a Contract by calling Update() and setting the value in its Status field to Activated; however, the Status field is the only field that the client application can set in the Update() call when activating the Contract.

Once a Contract has been activated, your client application cannot change its status; however, prior to activation, your client application can change the status value from Draft to InApproval via the API. Also, your client application can delete contracts whose status is Draft or InApproval but not when a contract status is Activated.

Client applications can use the API to Create(), Update(), and Query() any Attachment associated with a contract.

ContractContactRole

Represents the role that a given Contact plays on a Contract.

Corresponds to an SObject4 in which the ObjectType="ContractContactRole".

Supported Calls

Create(),Update(),DescribeSObjects(),Query(),Retrieve(),CreateObject(),GetDeleted(), GetUpdated()

Field	Field Type	Field Properties	Description
ContactId	reference	Create Filter Update	ID of the Contact associated with this Contract. For information on IDs, see ID Field Type.
ContractId	reference	Create Filter	Required. ID of the Contract. For information on IDs, see ID Field Type.
IsDeleted	boolean	Defaulted on create Filter	Indicates whether the object has been moved to the Recycle Bin (true) or not (false). Label is Deleted .
IsPrimary	boolean	Defaulted on create Filter Update	Specifies whether this Contact plays the primary role on this Contract (true) or not (false). Note that each contract has only one primary contact role. Default is false. Labels is Primary .
Role	picklist	Create Filter Nillable Update	Name of the role played by the Contact on this Contract, such as Decision Maker, Approver, Buyer, and so on. Must be unique—there cannot be multiple records in which the ContractId, ContactId, and Role values are identical. Different contacts can play the same role on the same contract. A contact can play different roles on the same contract.

Use the ContractContactRole object to define the role that a given Contact plays on a given Contract within the context of a specific Opportunity.

ContractHistory

Represents the history of changes to the values in the fields of a contract.

Supported Calls

```
Query(),Retrieve(),GetDeleted(),GetUpdated(),and DescribeSObjects()
```

Fields

Field	Field Type	Field Properties	Description
ContractId	reference	Filter	ID of the Contract. Label is Contract ID. For information on IDs, see ID Field Type.
Field	picklist	Filter Restricted picklist	The name of the field that was changed.
IsDeleted	boolean	Defaulted on create Filter	Indicates whether the object has been moved to the Recycle Bin (true) or not (false). Label is Deleted .

Usage

Use this object to identify changes to a contract.

This object respects field level security on the parent object.

ContractStatus

Represents the status of a Contract, such as Draft, InApproval, Activated, Terminated, or Expired.

Corresponds to an SObject4 in which the ObjectType="ContractStatus".

```
Query(),Retrieve(),CreateObject(),and DescribeSObjects()
```

Field	Field Type	Field Properties	Description
IsDefault	boolean	Defaulted on create	Indicates whether this is the default contract status value (true) or not (false) in the picklist.
		Filter	
MasterLabel string	string	Filter	Master label for this contract status value. This display value
		Nillable	is the internal label that does not get translated.
SortOrder	int	Filter	Number used to sort this value in the contract status picklist.
		Nillable	previous contract status values might have been deleted.
StatusCode	picklist	Filter	Code indicating the status of a contract. One of the following
		Nillable	• Draft
Restricted	Restricted	• InApproval	
		picklist	• Activated
			Two other values (Terminated and Expired) are defined but are not available for use via the API.

Usage

This object represents a value in the contract status picklist. The contract status picklist provides additional information about the status of a Contract, such as its current state (Draft, InApproval, or Activated). Your client application can invoke the Query () call on the this object to retrieve the set of values in the contract status picklist, and then use that information while processing Contract objects to determine more information about a given contract. For example, the application could test whether a given contract is activated based on its Status value and the value of the StatusCode property in the associated ContractStatus object.

This object is read-only via the API. With sufficient permissions, your client application can invoke the Query() and DescribeSObjects() calls on it.

ContractTag

Associates a word or short phrase with a Contract.

```
Create(),Query(),Retrieve(),DescribeSObjects()
```

Field	Field Type	Field Properties	Description
ItemId	reference	Create Filter	ID of the tagged item.
Name	string	Create Filter	Name of the tag. If this value does not already exist, a new TagDefinition is created and becomes the parent of this Tag object. Otherwise, a TagDefinition with the same name becomes the parent of this Tag object. Parent relationships are created automatically.
TagDefinitionId	reference	Filter	ID of the parent TagDefinition object that owns the tag.
Туре	picklist	Create Filter Restricted picklist	 Defines the visibility of a tag. Possible value are: Public: The tag can be viewed and manipulated by all users in an organization Personal: The tag can be viewed or manipulated only by a user with a matching OwnerId

Usage

ContractTag stores the relationship between its parent TagDefinition and the Contract being tagged. Tag objects act as metadata, allowing users to describe and organize their data.

When a tag is deleted, its parent TagDefinition will also be deleted if the name is not being used; otherwise, the parent remains. Deleting a TagDefinition sends it to the recycle bin, along with any associated tag entries.

For more information on tags, see "About Tagging" in the Salesforce online help.

CurrencyType

Represents the currencies used by an organization for which the multicurrency feature is enabled.

Corresponds to an SObject4 in which the ObjectType="CurrencyType".

Supported Calls

```
Create(),Update(),Query(),Search(),Retrieve(),GetUpdated(),CreateObject(),
DescribeSObjects()
```

Special Access Rules

- · This object is not available in single-currency organizations.
- You need the "Customize Application" permission to edit this object.
- Your client application cannot delete this object.
- Customer Portal users cannot access this object.

Field	Field Type	Field Properties	Description
ConversionRate	double	Filter	Required. Conversion rate of this currency type against the corporate currency.
DecimalPlaces	int	Filter	Required. For this currency, specifies the number of digits to the right of the decimal point, such as zero (0) for JPY or 2 for USD. Required.
IsActive	boolean	Defaulted on create Filter	Indicates whether this currency type is active (true) or not (false). Inactive currency types do not appear in picklists in the Salesforce user interface. Label is Active .
IsCorporate	boolean	Defaulted on create Filter	Indicates whether this currency type is the corporate currency (true) or not (false). Label is Corporate Currency . All other currency conversion rates are applied against this corporate currency. If a currency is already defined as the corporate currency in the Salesforce user interface, it cannot be unset. When a non-corporate currency is set to a corporate currency, the system will reconfigure all conversion rates based on the new corporate currency.
IsoCode	picklist	Filter Restricted picklist	Required. ISO code of the currency. Required. Must be one of the valid alphabetic, three-letter currency ISO codes defined by the ISO 4217 standard, such as USD, GBP, or JPY. Must be unique within your organization. Label is Currency ISO Code .

Usage

For multicurrency organizations only, use the CurrencyType object to define the currencies that your organization uses.

DatedConversionRate

Represents the dated exchange rates used by an organization for which the multicurrency and the effective dated currency features are enabled.

Supported Calls

Update(),Query(),Retrieve(),GetDeleted(),GetUpdated(),CreateObject(),DescribeSObjects()

Special Access Rules

Customer Portal users cannot access this object.

Field	Field Type	Field Properties	Description
ConversionRate	double	Filter Update	Required. Conversion rate of this currency type against the corporate currency.
IsoCode	picklist	Filter Restricted picklist	Required. ISO code of the currency. Required. Must be one of the valid alphabetic, three-letter currency ISO codes defined by the ISO 4217 standard, such as USD, GBP, or JPY. Must be unique within your organization. Label is Currency ISO Code .
NextStartDate	date	Filter Nillable	Read only. The date on which the next effective dated exchange rate will start. Effectively the day after the end date for this exchange rate.
StartDate	date	Filter	The date on which the effective dated exchange rate will start.

Usage

For multicurrency organizations with advanced currency management enabled, use the DatedConversionRate object to define the exchange rates that your organization uses for a date range. This object is not available in single-currency organizations, nor is it available if the organization does not have advanced currency management enabled.

Division

A logical segment of your organization's data. For example, if your company is organized into different business units, you could create a division for each business unit, such as "North America," "Healthcare," or "Consulting."

Supported Calls

```
Create(),Update(),Query(),Retrieve(),GetUpdated(),DescribeSObjects()
```

Special Access Rules

- Divisions must be enabled for your organization to access this object. For instructions to determine whether divisions have been enabled, see the section "Usage" in this topic.
- Customer Portal users cannot access this object.

Field	Field Type	Field Properties	Description
IsActive boolean	boolean	Create Defaulted on create	Indicates whether the division is active (true) or not (false). Label is Active .
	Filter		
		Update	

Field	Field Type	Field Properties	Description
IsGlobalDivision	boolean	Defaulted on create Filter	Indicates whether the division is your organization's global default division (true) or not (false). Label is Global Division .
Name	string	Create Filter Update	A descriptive name for the division. Limit: 80 characters.
SortOrder	int	Create Filter Nillable Update	The order in which this division name appears in the Division picklist field when creating or editing users in the Salesforce user interface.

To discover whether divisions have been enabled for an organization, inspect the User or Group object for the DefaultDivision field--if it is present, then divisions have been enabled, and this field (the field is named Division in objects other than User and Group) will be available in all relevant objects.

The values available for that field are the global division ID for the organization, created when divisions are first enabled, and any other division IDs that have been created. The division ID associated with a user is populated in the objects owned or created by the user.

You can use the division ID to make searches, reports, and list views run more quickly and return more relevant results if an organization has very large data sets. For more information, see the Salesforce online help, in the Fields description for the object.

You can use WITH in SOSL to pre-filter results based on division. This is faster than specifying the division in a WHERE clause. For more information, see WITH *DivisionFilter*.

For more information about how divisions are used, see the topic "Overview of Divisions" in the Salesforce online help.



Note: The User object has a Division field that is unrelated to this object. The Division field is a standard text field similar to Company or Department that has no special properties. Do not confuse it with the DefaultDivision field, which does relate to this object.

Document

Represents a file that a user has uploaded. Unlike Attachment objects, Documents are not attached to a parent object.

Corresponds to an SObject4 in which the ObjectType="Document".

```
Create(),Update(),DescribeSObjects(),Query(),Search(),Retrieve(),CreateObject(),
GetDeleted(),GetUpdated()
```

Special Access Rules

You must have the "Edit" permission on documents and the appropriate access to the Folder that contains a document in order to create or update a document in that Folder.

Field	Field Type	Field Properties	Description
AuthorID	reference	Create	ID of the User who is responsible for the Document. For
		Defaulted on create	information on IDs, see ID Field Type.
		Filter	
		Update	
Body	base64	Create	Required. Encoded file data. If specified, then do not specify
		Nillable	a URL.
		Update	Note: Client applications are responsible for the conversion of Base64 data between binary and String formats.
BodyLength	int	Filter	Size of the file (in bytes).
ContentType	string	Create	Type of content. Label is Mime Type . Limit: 120 characters.
		Filter	If the Disallow HTML documents and attachments
		Nillable	security setting is enabled for your organization, you cannot upload files with the following file extensions: htm. html
		Update	htt, htx, mhtm, mhtml, shtm, shtml, acgi.
Description	textarea	Create	Text description of the Document. Limit: 255 characters.
		Filter	
		Nillable	
		Update	
DeveloperName	string	Create	The unique name of the object in the API. The name can
		Filter	contain only alphanumeric characters and must begin with a letter. In managed packages, this field prevents naming
		Nillable	conflicts on package installations. With this field, a developer
		Update	can change the object's name in a managed package and the changes are reflected in a subscriber's organization. Label is Document Unique Name .
FolderId	reference	Create	Required. ID of the Folder that contains the Document. See
		Filter	Folder.
		Update	

Field	Field Type	Field Properties	Description
IsBodySearchable	boolean	Defaulted on create Filter	Indicates whether the contents of the object can be searched using a SOSL FIND call. The ALL FIELDS search group includes the content as a searchable field.
IsDeleted	boolean	Defaulted on create Filter	Indicates whether the object has been moved to the Recycle Bin (true) or not (false). Label is Deleted .
IsInternalUseOnly	boolean	Create Defaulted on create Filter Update	Indicates whether the object is only available for internal use (true) or not (false). Label is Internal Use Only .
IsPublic	boolean	Create Defaulted on create Filter Update	Indicates whether the object is available for external use (true) or not (false). Label is Externally Available .
Keywords	string	Create Filter Nillable Update	Keywords. Limit: 255 characters.
Name	string	Create Filter Update	Required. Name of the document. Label is Document Name .
Туре	Create Filter Nillable Update string	Create Filter Nillable Update	File type of the Document. In general, the values match the file extension for the type of Document (such as pdf or jpg). Label is File Extension .
URL	string	Create Filter Nillable Update	URL reference to the file (instead of storing it in the database). If specified, do not specify the Body or BodyLength.

When calling Create() or Update() for a document, a client application can specify a value in either the Body or URL fields, but not both.

Encoded Data

The API sends and receives the binary file data encoded as a base64 data type. Prior to Create(), clients must encode the binary file data as base64. Upon receiving an API response, clients must decode the base64 data to binary.

Maximum Document Size

The Create() and Update() calls restrict documents to a maximum size of 5 MB.

DocumentAttachmentMap

Maps the relationship between an EmailTemplate and its attachment, which is stored as a Document.

Supported Calls

Create(),Query(),Retrieve(),Update()

Special Access Rules

Customer Portal users cannot access this object.

Fields

Field	Field Type	Field Properties	Description
DocumentId	reference	Create Filter Update	ID of the document that this object tracks.
DocumentSequence	int	Create Filter Update	Represents the order that the attachments will be included in the email defined by the EmailTemplate specified by the DocumentId. Label is Attachment Sequence. The first attachment is given a value of 0, and each subsequent attachment is given a value incremented by 1.
ParentId	reference	Create Filter Update	ID of the EmailTemplate parent. The attachment identified by DocumentId is attached to the EmailTemplate specified in this field.

Usage

Use this object to map the relationship of an EmailTemplate to its attachments, and to specify the order of the attachments.

DocumentTag

Associates a word or short phrase with a Document.

Supported Calls

Create(),Query(),Retrieve(),DescribeSObjects()

Fields

Field	Field Type	Field Properties	Description
ItemId	reference	Create Filter	ID of the tagged item.
Name	string	Create Filter	Name of the tag. If this value does not already exist, a new TagDefinition is created and becomes the parent of this Tag object. Otherwise, a TagDefinition with the same name becomes the parent of this Tag object. Parent relationships are created automatically.
TagDefinitionId	reference	Filter	ID of the parent TagDefinition object that owns the tag.
Туре	picklist	Create Filter Restricted picklist	 Defines the visibility of a tag. Possible value are: Public: The tag can be viewed and manipulated by all users in an organization Personal: The tag can be viewed or manipulated only by a user with a matching OwnerId

Usage

Document Tag stores the relationship between its parent TagDefinition and the Document being tagged. Tag objects act as metadata, allowing users to describe and organize their data.

When a tag is deleted, its parent TagDefinition will also be deleted if the name is not being used; otherwise, the parent remains. Deleting a TagDefinition sends it to the recycle bin, along with any associated tag entries.

For more information on tags, see "About Tagging" in the Salesforce online help.

EmailMessage

An email message.

Supported Calls

Create(),DescribeSObjects(),Query(),Retrieve(),GetDeleted(),GetUpdated()

Special Access Rules

Customer Portal users have read access to this object.

Field	Field Type	Field Properties	Description
ActivityId	reference	Create	ID of the activity that is associated with the email. Usually
		Filter	represents an open task that is created for the case owner when a new unread email message is received. For
		Nillable	information on IDs, see ID Field Type.
BccAddress	string	Create	The addresses that were sent a blind carbon copy of the email.
		Filter	
		Nillable	
CcAddress	string	Create	The addresses that were sent a carbon copy of the email.
		Filter	
		Nillable	
FromAddress	email	Create	The address that originated the email.
		Filter	
		Nillable	
FromName	string	Create	The sender's name.
		Filter	
		Nillable	
HasAttachment	boolean	Defaulted on create	Indicates whether the email was sent with an attachment (true) or not (false).
		Filter	
Headers	textarea	Create	The Internet message headers of the incoming email. Used
		Nillable	for debugging and tracing purposes. Does not apply to outgoing emails.
HtmlBody	textarea	Create	The body of the email in HTML format.
		Nillable	
Incoming	boolean	Create	Indicates whether the email was received (true) or sent
		Defaulted on create	(Ialse).
		Filter	
IsDeleted	boolean	Defaulted on create	Indicates whether the object has been moved to the Recycle Bin (true) or not (false). Label is Deleted .
		Filter	

Field	Field Type	Field Properties	Description
MessageDate	dateTime	Create	The date the email was created.
		Filter	
		Iniliable	
ParentId	ID	Create	ID of the Case to which the email is associated.
		Filter	
Status	picklist	Create	Read only. The status of the email. For example, "New,"
		Filter	"Unread, "Replied," Sent."
		Restricted	
		picklist	
Subject	string	Create	The subject line of the email.
		Filter	
		Nillable	
TextBody	textarea	Create	The body of the email, in plain text format.
		Nillable	
ToAddress	string	Create	The address of the email's recipient.
		Filter	
		Nillable	

This object supports the Email-to-Case feature, which allows email sent to one of your company's email addresses to automatically create a new case in Salesforce. See "What are Email Services?" in the Salesforce online help for more information.

EmailServicesAddress

An email service address.

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.

```
Create(),Update(),DescribeSObjects(),Query(),Retrieve(),GetDeleted(),GetUpdated()
```

Field	Field Type	Field Properties	Description
AuthorizedSenders	textarea	Create Filter Nillable Update	Configures the email service address to only accept messages from the email addresses and/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.
EmailDomainName	string	Create Filter Nillable	A read only field you can query that contains the Salesforce-generated domain part of this email service address. Salesforce generates a unique domain-part for each email service address to ensure that no two email service addresses are identical.
FunctionId	reference	Create	The ID of the email service for which the email service address receives messages.
IsActive	boolean	Create Defaulted on create Filter Update	Indicates whether this object is active (true) or not (false).
LocalPart	string	Create Filter	The local-part of the email service address. For the local-part of a Salesforce email address, all alphanumeric characters are valid, plus the following special characters: !#\$%&:'/=?^_+-`{ }~. The dot character (.) is also valid as long as it is not the first or last character. Salesforce email addresses are case-insensitive.
RunAsUserId	reference	Defaulted on create Filter	The user whose permissions the email service assumes when processing messages sent to this address.

Usage

This object supports the email services feature, which allows you to create automated processes that use Apex classes to process the contents, headers, and attachments of inbound email. For example, you can create an email service that automatically creates contact records based on contact information in messages. See "What are Email Services?" in the Salesforce online help for more information.

EmailServicesFunction

An email service.

Supported Calls

Create(),Update(),DescribeSObjects(),Query(),Retrieve(),GetDeleted(),GetUpdated()

Field	Field Type	Field Properties	Description
AddressInactiveAction	string	Create Filter Nillable Restricted picklist Update	 Indicates what the email service does with messages received at an email address that is inactive. One of the following values: 0—The system default is used. 1—The email service returns the message to the sender with a notification that explains why the message was rejected. 2—The email service deletes the message without notifying the sender. 3—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.
ApexClassId	ID	Create Filter Nillable Update	The ID of the Apex class that the email service uses to process inbound messages.
AttachmentOption	string	Create Filter Restricted picklist Update	 Indicates the types of attachments the email service accepts. One of the following values: 0—The email service accepts the message but discards any attachment. 1—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. 2—The email service only accepts binary attachments, such as image, audio, application, and video files. 3—The email service accepts any type of attachment.

Field	Field Type	Field Properties	Description
AuthenticationFailureAction	string	Create Filter Nillable Restricted picklist Update	 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: 0—The system default is used. 1—The email service returns the message to the sender with a notification that explains why the message was rejected. 2—The email service deletes the message without notifying the sender. 3—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	ID	Create Filter Nillable Restricted picklist Update	 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: 0—The system default is used. 1—The email service returns the message to the sender with a notification that explains why the message was rejected. 2—The email service deletes the message without notifying the sender. 3—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	textarea	Create Filter Nillable Update	Configures the email service to only accept messages from the email addresses and/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.
FunctionInactiveAction	string	Create Filter Nillable Restricted picklist	 Indicates what the email service does with messages it receives when the email service itself is inactive. One of the following values: 0—The system default is used.

Field	Field Type	Field Properties	Description
		Update	 1—The email service returns the message to the sender with a notification that explains why the message was rejected. 2—The email service deletes the message without notifying the sender. 3—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	Create	The name of the email service.
		Filter	
		Update	
IsActive	boolean	Create	Indicates whether this object is active (true) or not (false).
		Defaulted on create	
		Filter	
		Update	
IsAuthenticationRequired	boolean	Create	Configures the email service to verify the legitimacy of the
Defa on cr	Defaulted on create	uses the SPF, SenderId, and DomainKeys protocols to verify the sender's legitimacy: If the sending server passes at least	
		Filter	one of these protocols and does not fail any, the email service
		Update	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.
IsTlsRequired	boolean	Create	Configures the email service to use Transport Layer Security
		Defaulted on create	(1LS), a protocol for secure email communication, to ensure the security and authenticity of inbound email.
		Filter	
		Update	
OverLimitAction	string	Create	Indicates what the email service does with messages if the
		Filter	total number of messages processed by all email services combined has reached the daily limit for your organization.
		Nillable	One of the following values:
		Restricted	• 0—The system default is used.
		Update	• 1—The email service returns the message to the sender with a notification that explains why the message was rejected.

Field	Field Type	Field Properties	Description
			 2—The email service deletes the message without notifying the sender. 3—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.
			Salesforce calculates the limit by multiplying the number of user licenses by 1,000.

This object supports the email services feature, which allows you to create automated processes that use Apex classes to process the contents, headers, and attachments of inbound email. For example, you can create an email service that automatically creates contact records based on contact information in messages. See the Salesforce online help for more information.

EmailStatus

Represents the status of email sent.

Supported Calls

DescribeGlobal(),DescribeSObject(),DescribeSObjects()

Special Access Rules

Customer Portal users cannot access this object.

Field	Field Type	Field Properties	Description
EmailTemplateName	string	Filter Nillable	The name of the EmailTemplate.
FirstOpenDate	dateTime	Filter Nillable	Date when the email was first opened by recipient. Label is Date Opened .
TaskId	reference	Filter	The activity (task or event) associated with the email. Label is Activity ID.
TimesOpened	int	Filter	Number of times the recipient opened the email.
WhoId	reference	Filter Nillable	The contact or lead ID associated with the recipient. Label is Contact/Lead ID .

EmailTemplate

Represents a template for mass email, or email sent when the activity history related list of a record is modified.



Note: You cannot send a mass email using a Visualforce email template.

Supported Calls

Create(),Query(),Retrieve(),Update(),DescribeSObjects(),GetDeleted(),GetUpdated()

Special Access Rules

Customer Portal users cannot access this object.

Field	Field Type	Field Properties	Description
ApiVersion	double	Create Filter Nillable Update	The API version for this class. Every class has an API version specified at creation.
Body	textarea	Create	Content of the email.
		Nillable	
		Update	
BrandTemplateId	reference	Create	Required. ID of the BrandTemplate associated with this
		Filter	information for the email template.
		Nillable	1
Description	string	Create	Description of the template, for example, Promotion Mass
		Filter	Mailing.
		Nillable	
		Update	
DeveloperName	string	Create	The unique name of the object in the API. The name can
		Defaulted on create	contain only alphanumeric characters and must begin with a letter. In managed packages, this field prevents naming conflicts on package installations. With this field, a developer
		Filter	can change the object's name in a managed package and the
		Nillable	changes are reflected in a subscriber's organization. Label is
		Update	Template Offique Ivanie.
Encoding	picklist	Create	Character set encoding for the template.
		Filter	

Field	Field Type	Field Properties	Description
		Nillable	
		Restricted picklist	
		Update	
FolderId	reference	Create	ID of the folder that contains the template.
		Filter	
		Update	
HtmlValue	textarea	Create	This field contains the content of the email message,
		Nillable	including HTML coding to render the email message. Limit: 384 KB
		Update	
IsActive	boolean	Create	Indicates that this template is active if true, or inactive if
		Defaulted on create	false.
		Filter	
		Update	
LastUsedDate	dateTime	Filter	Date and time when this EmailTemplate was last used.
		Nillable	
Markup	textarea	Create	The Visualforce markup, HTML, Javascript, or any other
		Nillable	Web-enabled code that defines the content of the template.
		Update	
Name	string	Create	Name of the template. Label is Email Template Name .
		Filter	
		Nillable	
		Update	
NamespacePrefix	string	Create	The namespace prefix assigned to this object when it was
		Filter	created as part of a managed package. Null if this object is not part of a managed package. Limit is 15 characters. For
		Nillable	more information about managed packages and namespace prefixes, see "About Managed Packages" in the Salesforce online help.
			This field cannot be accessed unless the logged-in user has the "Customize Application" permission.
OwnerId	reference	Create	ID of the owner of the template.
		Filter	

Field	Field Type	Field Properties	Description
Subject	string	Create Nillable Update	Content of the subject line.
TemplateStyle	picklist	Create Filter Restricted picklist	Style of the template, such as formalLetter or freeform.
TemplateType	picklist	Create Filter Restricted picklist	Type of template, either HTML, text, or custom templates, or those generated byVisualforce.
TimesUsed	int	Filter Nillable	Number of times this template has been used.

To retrieve this object, issue a describe call on an object, which will return a query result for each activity since the object was created. You cannot use Query().

EntityHistory

Represents historical information about an object's changed field values. This object is only available to users with the "View All Data" permission.

This object is deprecated beginning with API version 8.0. Use the object-specific History objects instead: CaseHistory, ContractHistory, LeadHistory, OpportunityFieldHistory, OpportunityHistory, OrderHistory, ProcessInstanceHistory, QuantityForecastHistory, RevenueForecastHistory, or SolutionHistory.

Supported Calls

Query(),Retrieve(),GetUpdated(),GetDeleted(),DescribeSObjects()

Field	Field Type	Field Properties	Description
FieldName	picklist	Filter Restricted picklist	ID of the standard or custom field. For information on IDs, see ID Field Type.

Field	Field Type	Field Properties	Description
IsDeleted	boolean	Defaulted on create Filter	Indicates whether the object has been moved to the Recycle Bin (true) or not (false). Label is Deleted .
NewValue	anyType	Nillable	New value of the modified field.
OldValue	test	Nillable	Previous value of the modified field.
ParentId	reference	Filter	ID of the object that contains the field.
ParentSobjectType	picklist	Filter Restricted picklist	The kind of object that contains the field.

In API version 7.0 and later, this object works with Case, Contract, and Solution objects:

- This object is always read-only in the online application.
- When a field is modified, this object records both the old and new field values.

There are exceptions to this behavior for certain fields such as long text areas and multi-select picklists. These fields appear in this object to indicate that the field was changed, but the old and new values are not recorded.

- Two rows are added to this object when foreign key fields change. One row contains the foreign key object names that display in the online application. For example, "Jane Doe" is recorded as the name of a contact. The other row contains the actual foreign key ID that is only returned to and visible from the API.
- Up to a total of twenty fields (standard or custom) can be tracked for a given object.
- In the online application, you can specify which fields are tracked or not tracked at any time. See the Salesforce online help for more information.
- As soon as tracking is turned on for a field, all changes to its value are recorded in the database.
- Turning off tracking for a field stops further changes from being recorded, but the history data is not deleted.
- Be advised that deleting a custom field also permanently deletes the history data for that custom field.

Event

Represents a calendar appointment event.

Corresponds to an SObject4 in which the ObjectType="Event".

```
Create(),Update(),Query(),Search(),Retrieve(),CreateObject(),GetDeleted(),GetUpdated(),
DescribeSObjects()
```

Field	Field Type	Field Properties	Description
AccountId	reference	Filter Nillable	ID of the related Account.
ActivityDate	date	Create Filter Nillable Update	If the Event IsAllDayEvent flag is set to true (indicating that it is an all day Event), then the Event due date information is contained in the ActivityDate field. This field is a date field with a timestamp that is always set to midnight in the Coordinated Universal Time (UTC) time zone. The timestamp is not relevant, and you should not attempt to alter it to account for any time zone differences. Label is Due Date Only .
			This field is required in version 12.0 and earlier if the ISAllDayEvent flag is set to true.
			The value for this field and StartDateTime must match, or one of them must be null.
ActivityDateTime	dateTime	Create Filter Nillable Update	If the Event IsAllDayEvent flag is set to false (indicating that it is not an all day Event), then the Event due date information is contained in the ActivityDateTime field. This field is a regular Date/Time field with a relevant time portion. The time portion is always transferred in the Coordinated Universal Time (UTC) time zone. You need to translate the time portion to or from a local time zone for the user or the application, as appropriate. Label is Due Date Time .
			This field is required in version 12.0 and earlier if the IsAllDayEvent flag is set to false.
			The value for this field and StartDateTime must match, or one of them must be null.
Description	textarea	Create Nillable Update	Text description of the event. Limit: 32,000 characters.
DurationInMinutes	int	Create Filter Nillable	The event length, in minutes. Note that even though this represents a temporal value, it is an integer type—not a dateTime type. This field is required in version 12.0 and earlier if
		Update	IsAllDayEvent is false. In version 13.0 and later, this field is optional based on the following:
			• If IsAllDayEvent is true, you can supply a value for either DurationInMinutes or EndDateTime. Supplying values in both fields is allowed if they evaluate

Field	Field Type	Field Properties	Description
			 to the same span of time. If both fields are null, the duration defaults to one day. If IsAllDayEvent is false, a value must be supplied for either DurationInMinutes or EndDateTime. Supplying values in both fields is allowed if they evaluate to the same span of time. If the multiday event feature is enabled in Salesforce, then API versions 13.0 and later support values greater than 1440 for the DurationInMinutes field. API versions 12.0 and earlier cannot access event objects whose DurationInMinutes is greater than 1440. For more information, see Multiday Events.
EndDateTime	dateTime	Create Filter Nillable Update	 Available in version 13.0 and later. This field is optional based on the following: If IsAllDayEvent is true, you can supply a value for either DurationInMinutes or EndDateTime. Supplying values in both fields is allowed if they evaluate to the same span of time. If both fields are null, the duration defaults to one day. If IsAllDayEvent is false, a value must be supplied for either DurationInMinutes or EndDateTime. Supplying values in both fields is allowed if they evaluate to the same span of time.
IsAllDayEvent	boolean	Create Defaulted on create Filter Update	Indicates whether the ActivityDate field (true) or the ActivityDateTime field (false) is used to define the date and/or time of the event. Label is All Day Event . See also DurationInMinutes and EndDateTime.
IsChild	boolean	Defaulted on create Filter	Indicates whether the event is a child of another event (true) or not (false).
IsDeleted	boolean	Defaulted on create Filter	Indicates whether the object has been moved to the Recycle Bin (true) or not (false). Label is Deleted .
IsGroupEvent	boolean	Defaulted on create Filter	Indicates whether the event is a group event with multiple attendees (true) or not (false).
IsPrivate	boolean	Create Defaulted on create	Indicates whether users other than the creator of this event can (false) or cannot (true) see the event details when viewing the event user's calendar. However, users with the "View All Data" or "Modify All Data" permission can see

Field	Field Type	Field Properties	Description
		Filter Update	private events in reports and searches, or when viewing other users' calendars. Private events cannot be associated with opportunities, accounts, cases, campaigns, contracts, leads, or contacts. Label is Private .
IsRecurrence	boolean	Create Defaulted on create Filter	Indicates whether the event is scheduled to repeat itself (true) or only occurs once (false). This is a read only field on update, but not on create. If this field value is true, then RecurrenceEndDateOnly, RecurrenceStartDateTime, RecurrenceType, and any recurrence fields associated with the given recurrence type must be populated. See Recurring Events. Label is Create recurring series of events.
IsReminderSet	boolean	Create Defaulted on create Filter Update	Indicates whether the event is a reminder (true) or not (false).
IsVisibleInSelfService	boolean	Defaulted on create Filter	Indicates whether the event can be viewed in the Customer Self-Service Portal (true) or not (false).
Location	string	Create Filter Nillable Update	The location of the event.
OwnerId	reference	Create Defaulted on create Filter Update	ID of the user who owns the event. Label is Assigned to ID .
RecurrenceActivityId	reference	Filter Nillable	Read-only. Not required on create. ID of the main record of the recurring event. Subsequent occurrences have the same value in this field.
RecurrenceDayOfMonth	int	Create Filter Nillable Update	The day of the month on which the event repeats.

Field	Field Type	Field Properties	Description
RecurrenceDayOfWeekMask	int	Create Filter Nillable Update	 The day or days of the week on which the event repeats. This field contains a bitmask. For each day of the week, the values are as follows: Sunday = 1 Monday = 2 Tuesday = 4 Wednesday = 8 Thursday = 16 Friday = 32 Saturday = 64 Multiple days are represented as the sum of their numerical values. For example, Tuesday and Thursday = 4 + 16 = 20.
RecurrenceEndDateOnly	date	Create Filter Nillable Update	The last date on which the event repeats. For multiday recurring events, this is the day on which the last occurrence starts.
RecurrenceInstance	picklist	Create Filter Nillable Restricted picklist Update	The frequency of the recurring event. For example, "2nd" or "3rd."
RecurrenceInterval	int	Create Filter Nillable Update	The interval between recurring events.
RecurrenceMonthOfYear	picklist	Create Filter Nillable Restricted picklist Update	The month of the year on which the event repeats.
RecurrenceStartDateTime	dateTime	Create Filter Nillable Update	The date and time when the recurring event begins. Must be a date and time before RecurrenceEndDateOnly.
Field	Field Type	Field Properties	Description
--------------------------	---------------	------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
RecurrenceTimeZoneSidKey	picklist	Create Filter	The time zone associated with the recurring event. For example, "UTC-8:00" for Pacific Standard Time.
		Nillable	
		Restricted picklist	
		Update	
RecurrenceType	picklist	Create	Indicates how often the event repeats. For example, daily,
	1	Filter	weekly, or every Nth month (where "Nth" is defined in
		Nillable	Recurrenceinstance).
		Restricted picklist	
		Update	
ReminderDateTime	dateTime	Create	Represents the time the reminder is scheduled to fire, if
		Filter	IsReminderSet is set to true. If it is set to false, then
		Nillable	Salesforce user interface, or the reminder has already fired at
		Update	the time indicated by the value.
ShowAs	picklist	Create	Indicates how this event appears when another user views the calendar: Busy, Out of Office, or Free Time. Label is Show Time As .
		Filter	
		Nillable	
		Restricted picklist	
		Update	
StartDateTime	dateTime	Create	Available in version 13.0 and later. The start date and tim
		Filter	of the event.
		Nillable	If the Event IsAllDayEvent flag is set to true (indicating that it is an all day Event) then the Event start date
		Update	information is contained in the StartDateTime field. This field is a date field with a timestamp that is always set to midnight in the Coordinated Universal Time (UTC) time zone. The timestamp is not relevant, and you should not attempt to alter it to account for any time zone differences. If the Event IsAllDayEvent flag is set to false (indicating that it is not an all day Event), then the Event start date information is contained in the StartDateTime field. The time portion is always transferred in the Coordinated Universal Time (UTC) time zone. You need to translate the time portion to or from a local time zone for the user or the
			application, as appropriate.

Field	Field Type	Field Properties	Description
			If this field has a value, then ActivityDate and ActivityDateTime must either be null or match the value of this field.
Subject	comobobox	Create	The subject line of the event, such as Call, Email, or Meeting.
		Filter	
		Nillable	
		Update	
Туре	picklist	Create	The type of event, such as Call, Email, or Meeting.
		Filter	
		Nillable	
		Update	
WhatId	reference	Create	ID of a related Account, Opportunity, Campaign, Case, or
		Filter	custom object. Label is Opportunity/Account ID .
		Nillable	
		Update	
WhoId	reference	Create	ID of a related Contact or Lead. If the Whold refers to a lead,
		Filter	then the WhatId field must be empty. Label is Contact/Lead
		Nillable	
		Update	

Use Events to manage calendar appointments.

Queries on events will be denied before they time out if they involve amounts of data that are deemed too large. In such cases, MALFORMED_QUERY is returned. If you receive the MALFORMED_QUERY error message, refactor your query to return or scan a smaller amount of data.

When querying for events with a specific due date, you must filter on both the ActivityDateTime and ActivityDate fields. For example to find all events with a due date of February 14, 2003, you need two filters:

- One filter with the ActivityDate field equal to the Coordinated Universal Time (UTC) time zone on February 14, 2003.
- One filter with the ActivityDateTime field greater than or equal to midnight on February 14, 2003 in the user's local time zone AND less than or equal to midnight on February 15, 2003 in the user's local time zone.

Alternatively, in version 13.0 and later, you can find events with a specific due date by filtering on StartDateTime. For example, to find all events with a due date of February 14, 2003, filter with the StartDateTime greater than or equal to midnight on February 14, 2003 in the user's local time zone AND less than or equal to midnight on February 15, 2003 in the user's local time zone.

Multiday Events

- Multiday events are available in version 13.0 and later. Also, in earlier versions SOQL queries do not return multiday events.
- Multiday events are enabled through the user interface at Setup ➤ Customize ➤ Activities ➤ Activity Settings.
- If the multiday event feature is enabled in Salesforce, then API versions 13.0 and later support values greater than 1440 for the DurationInMinutes field. API versions 12.0 and earlier cannot access event objects whose DurationInMinutes is greater than 1440.
- Multiday events cannot exceed 14 days.

Recurring Events

- Recurring events are available in version 7.0 and later.
- After an event is created, it cannot be changed from recurring to nonrecurring or vice versa.
- When creating a recurring event series, the duration of the event must be 24 hours or less (either the DurationInMinutes or the difference between RecurrenceStartDateTime and EndDateTime cannot be greater than 24 hours). Once the recurring event series is created, you can extend the length of individual occurrences beyond 24 hours if Multiday events are enabled; see Multiday Events.
- If IsRecurrence is true, then RecurrenceStartDateTime, RecurrenceEndDateOnly, RecurrenceType, and any properties associated with the given recurrence type (see the following table) must be populated.

The following table describes the usage of recurrence fields. Each recurrence type must have all of its properties set. All unused properties must be set to null.

Recurrence Type Value	Properties	Example Pattern
RecursDaily	RecurrenceInterval	Every second day
RecursEveryWeekday	RecurrenceDayOfWeekMask	Every weekday - cannot be Saturday or Sunday
RecursMonthly	RecurrenceDayOfMonth RecurrenceInterval	Every second month, on the third day of the month
RecursMonthlyNth	RecurrenceInterval RecurrenceInstance RecurrenceDayOfWeekMask	Every second month, on the last Friday of the month
RecursWeekly	RecurrenceInterval RecurrenceDayOfWeekMask	Every three weeks on Wednesday and Friday
RecursYearly	RecurrenceDayOfMonth RecurrenceMonthOfYear	Every March on the twenty-sixth day of the month
RecursYearlyNth	RecurrenceDayOfWeekMask RecurrenceInstanceRecurrenceMonthOfYear	The first Saturday in every October

For information about working with archived events, see Archived Activities.

EventAttendee

Represents a person (User, Contact, or Lead) who has been invited to attend an Event, or a scheduled resource (such as a conference room) associated with the event.

Corresponds to an SObject4 in which the ObjectType="EventAttendee".

Supported Calls

Query(),Retrieve(),CreateObject(),GetUpdated(),GetDeleted(),DescribeSObjects()

Fields

Field	Field Type	Field Properties	Description
AttendeeId	reference	Filter	ID of the person (User, Contact, or Lead) invited to attend the Event, or the scheduled resource associated with this Event. For information on IDs, see ID Field Type. You cannot created an attendee who is the owner of the event. This restrictions prevents errors such as the event showing up twice on the owner's calendar.
EventId	reference	Filter	ID of the Event.
IsDeleted	boolean	Defaulted on create Filter	Indicates whether the object has been moved to the Recycle Bin (true) or not (false). Label is Deleted .
RespondedDate	dateTime	Filter Nillable	Date and time when the attendee (invitee) responded.
Response	string	Filter Nillable	Optional text that the attendee entered when responding to the event request.
Status	picklist	Filter Restricted picklist	 Attendee status. One of the following values: New—Invitee has received the invitation but has not yet responded. Declined—Invitee has declined to attend the event. Accepted—Invitee has accepted the invitation to attend the event. Deleted—The invitation has been deleted. Maybe—Reserved for future use. This is not a valid value.

Usage

This object provides information about who has been invited to attend a particular event and their response to that invitation. A client application can, for example, Query () this object for a given event, iterate through the list, examine the status, and send email notifications to every person who accepted the invitation.

To determine all the events that a particular person is attending during a given time period (for example, next week), a client application could query the Event object for a given date range, iterate through the results and, for each event, query the EventAttendee object and determine whether the particular person (AttendeeId) has accepted an invitation to that event.

EventTag

Associates a word or short phrase with an Event.

Supported Calls

Create(),Query(),Retrieve(),DescribeSObjects()

Fields

Field	Field Type	Field Properties	Description
ItemId	reference	Create Filter	ID of the tagged item.
Name	string	Create Filter	Name of the tag. If this value does not already exist, a new TagDefinition is created and becomes the parent of this Tag object. Otherwise, a TagDefinition with the same name becomes the parent of this Tag object. Parent relationships are created automatically.
TagDefinitionId	reference	Filter	ID of the parent TagDefinition object that owns the tag.
Туре	picklist	Create Filter Restricted picklist	 Defines the visibility of a tag. Possible value are: Public: The tag can be viewed and manipulated by all users in an organization Personal: The tag can be viewed or manipulated only by a user with a matching OwnerId

Usage

Event Tag stores the relationship between its parent TagDefinition and the Event being tagged. Tag objects act as metadata, allowing users to describe and organize their data.

When a tag is deleted, its parent TagDefinition will also be deleted if the name is not being used; otherwise, the parent remains. Deleting a TagDefinition sends it to the recycle bin, along with any associated tag entries.

For more information on tags, see "About Tagging" in the Salesforce online help.

FiscalYearSettings

Settings to define a custom or standard fiscal year for your organization. This object has a parent-child relationship with the Period object.

Supported Calls

```
Query(),Retrieve(),GetDeleted(),GetUpdated(),DescribeSObjects(),GetDeleted(),
GetUpdated()
```

Special Access Rules

Customer Portal users cannot access this object.

Fields

Field	Field Type	Field Properties	Description
IsStandardYear	boolean	Defaulted on create	Indicates whether the fiscal year is a standard calendar year (true) or a custom fiscal year (false).
		Filter	
Name	string	Filter	A name for the fiscal year. Limit: 80 characters.
PeriodId	reference	Filter	ID of the associated fiscal period. For information on IDs, see ID Field Type.
		Nillable	
PeriodLabelScheme	picklist	Filter	The numbering scheme used for fiscal periods.
		Nillable	
		Restricted picklist	
PeriodPrefix	picklist	Filter	The prefix of fiscal periods. For example, if "P" is the prefix,
		Nillable	then the first period is "P1."
QuarterLabelScheme	picklist	Filter	The numbering scheme used for fiscal quarters.
		Nillable	
		Restricted picklist	
QuarterPrefix	picklist	Filter	The prefix of fiscal quarters. For example, if "Q" is the prefix,
		Nillable	then the fourth quarter would be "Q4."
WeekLabelScheme	picklist	Filter	The numbering scheme used for weeks.
		Nillable	
		Restricted picklist	
YearType	picklist	Filter	Indicates one of two types of fiscal years, Standard or Custo
		Nillable	Standard denotes the standard Gregorian calendar, while
		Restricted picklist	Custom means a niscar year with a custom structure.

Usage

This is a read-only object. For information on setting up custom or standard fiscal years, see the Salesforce online help.

Folder

Represents a repository for a Document, MailMergeTemplate, EmailTemplate, or report. Only one type of item can be contained in a folder.

Corresponds to an SObject4 in which the ObjectType="Folder".

Supported Calls

```
Create(),Update(),DescribeSObjects(),Query(),Retrieve(),CreateObject(),GetDeleted(),
GetUpdated()
```

Special Access Rules

- You must have the "Modify All Data" permission to create, update, or delete document folders, email template folders, or report folders.
- To query Folders, no special permissions are needed.
- Customer Portal users cannot access this object.

Field	Field Type	Field Properties	Description
AccessType	picklist	Create Filter Restricted picklist Update	 Required. Indicates who can access the Folder. Available values include: Hidden—Folder is hidden from everyone. Public—Folder is accessible by all users. Shared—Folder is accessible only by Users in a particular Group or UserRole. The API does not allow you to view, insert, or update which group or Role the Folder is shared with.
DeveloperName	string	Create Filter Nillable Update	The unique name of the object in the API. The name can contain only alphanumeric characters and must begin with a letter. 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 Folder Unique Name .
IsReadOnly	boolean	Create Defaulted on create Filter Update	Indicates whether this Folder is read-only (true) or editable (false). Label is Read Only .
Name	string	Create Filter	Label of the folder as it appears in the Salesforce user interface. Label is Document Folder Label .

Field	Field Type	Field Properties	Description
		Nillable Update	
NamespacePrefix	string	Create Filter Nillable	The namespace prefix assigned to this object when it was created as part of a managed package. Null if this object is not part of a managed package. Limit is 15 characters. For more information about managed packages and namespace prefixes, see "About Managed Packages" in the Salesforce online help. This field cannot be accessed unless the logged-in user has the "Customize Application" permission.
Туре	picklist	Create Filter Restricted picklist	 Required. Type of objects contained in the Folder. This field cannot be updated. Available values include: Document Email template Report

Only one type of item can be contained in a folder, either Document, MailMergeTemplate, EmailTemplate, or report.

ForecastShare

Represents the sharing of a forecast at a given role and territory.

Corresponds to an SObject4 in which the ObjectType="ForecastShare".

Supported Calls

```
Query(),Retrieve(),DescribeSObjects()
```

Special Access Rules

Customer Portal users cannot access this object.

Field	Field Type	Field Properties	Description
AccessLevel	picklist	Filter Restricted picklist	A value (Read, Edit, or All) that represents the type of sharing being allowed.

Field	Field Type	Field Properties	Description
CanSubmit	boolean	Filter	Indicates whether the user or group can submit forecasts (True) or not (False).
RowCause	picklist	Filter Restricted picklist	 Reason that this sharing entry exists. Read-only. There are many possible values, including: Manual - The User or Group has access because a User has manually shared the forecast with them. Owner - The User is the owner of the forecast.
UserorGroupId	reference	Filter	The ID representing the User or Group being granted access.
UserRoleId	reference	Filter	ID of the UserRole associated with this object.

This object allows you to determine which users and groups can view and/or submit forecasts owned by other users. For more information, see Sharing in the Salesforce online help.

Group

Represents a set of Users.

Corresponds to an SObject4 in which the ObjectType="Group".

Supported Calls

```
Create(),Update(),DescribeSObjects(),Query(),Retrieve(),CreateObject(),GetDeleted(),
GetUpdated()
```

Special Access Rules

Customer Portal users cannot access this object.

Field	Field Type	Field Properties	Description
Email	email	Create	Email address for a group of type Case. Applies only for a case queue.
		Filter	
		Nillable	
		Update	
Name	string	Create	Required. Name of the group.
		Filter	
		Update	

Field	Field Type	Field Properties	Description
OwnerId	reference	Filter	ID of the user who owns the group.
RelatedId	reference	Filter Nillable	For Groups of type "Role," the ID of the associated UserRole. Read-only.
Туре	picklist	Create Filter Restricted picklist	 Required. Type of the Group. One of the following values: Regular—Standard public Group. When you Create() a Group, its type must be Regular, unless a PRM (Partner Relationship Management) Portal is enabled for the organization, in which case the type can be Regular or PRMOrganization. Role—Public Group that includes all of the Users in a particular UserRole. RoleAndSubordinates—Public Group that includes all of the Users in a particular UserRole. Organization—Public Group that includes all of the Users in the organization. This Group is read-only. Case—Public group of users, members of a queue that can own a Case. Lead—Public Group that includes all of the partners in an organization—Public Group that includes all of the partners in an organization—Public Group that includes all of the partners in an organization—Public Group that includes all of the partners in an organization—Public Group that includes all of the partners in an organization—Public Group that includes all of the partners in an organization—Public Group that includes all of the Users who are members of a queue. Queue—Public Group that includes all of the Users in an organization that has the PRM feature enabled. Queue—Public Group that includes all of the Users in an organization that has the PRM feature enabled. Territory—Public Group that includes all of the Users in an organization that has the territory feature enabled. TerritoryAndSubordinates—Public Group that includes all of the Users in a particular UserRole and all of the Users in UserRoles below that UserRole. Only Regular, Case, and Lead can be used when creating a group. The other values are reserved for salesforce.com use.

Unlike users, this object can be deleted. Any User can access this object-no special permissions are needed.

Only public groups are accessible via the API. Personal groups are not available.



Note: In API version 13.0 and later, if you delete a public Group, it will be deleted even if it has been used in sharing, consistent with the behavior for UserRoles. In versions before 13.0, such sharing would prevent the record from being deleted.

GroupMember

Represents a User or Group that is a member of a public group.

Corresponds to an SObject4 in which the ObjectType="GroupMember".

Supported Calls

```
Create(),DescribeSObjects(),Query(),Retrieve(),CreateObject(),GetDeleted(),GetUpdated()
```

Special Access Rules

Customer Portal users cannot access this object.

Fields

Field	Field Type	Field Properties	Description
GroupId	reference	Create Filter	Required. ID of the Group. For information on IDs, see ID Field Type.
UserOrGroupId	reference	Create Filter	Required. ID of the User or Group that is a direct member of the group. For information on IDs, see ID Field Type.

Usage

A record exists for every User or Group who is a direct member of a public group whose Type field is set to Regular. Users who are indirect members of Regular public groups are not listed as group members. A User can be an indirect member of a group if he or she is in a UserRole above the direct group member in the hierarchy, or if he or she is a member of a group that is included as a subgroup in that group.

If you attempt to create a record that matches an existing record, the Create() call simply returns the existing record.

Holiday

Represents a period of time during which your customer support team is unavailable. Business hours and escalation rules associated with business hours are suspended during any holidays with which they are affiliated.

Supported Calls

```
Create(),Update(),Query(),Retrieve()
```

Special Access Rules

Customer Portal users cannot access this object.

Field	Field Type	Field Properties	Description
ActivityDate	date	Create Filter Nillable Update	If the Holiday IsAllDay flag is set to true (indicating that it is an all-day holiday), then the holiday due date information is contained in the ActivityDate field. This field is a date field with a timestamp that is always set to midnight in the Coordinated Universal Time (UTC) time zone. The

Field	Field Type	Field Properties	Description
			timestamp is not relevant, and you should not attempt to alter it to account for any time zone differences.
Description	string	Create	Text description of the holiday.
		Filter	
		Nillable	
		Update	
EndTimeInMinutes	int	Create	The end time of the holiday in minutes.
		Filter	
		Nillable	
		Update	
IsAllDay	boolean	Create	Indicates whether the duration of the holiday is all day
		Defaulted on create	(true) or not (false).
		Filter	
		Update	
IsRecurrence	boolean	Create	Indicates whether the holiday is scheduled to repeat itself (true) or only occurs once (false). This is a read only field on update, but not on create. If this field value is true there are represented with the size of the
		Defaulted on create	
		Filter	type must be populated.
Name	string	Create	The name of the holiday.
		Filter	The nume of the honouy.
		Update	
RecurrenceDayOfMonth	int	Create	The day of the month on which the holiday repeats.
		Filter	
		Nillable	
		Update	
RecurrenceDayOfWeekMask	int	Create	The day or days of the week on which the holiday repeats.
		Filter	This field contains a bitmask. For each day of the week, the
		Nillable	 Sunday = 1
		Update	• Monday = 2
			• Tuesday = 4 Wednesday = 8
			 Thursday = 16
			• Friday = 32
			• Saturday = 64

Field	Field Type	Field Properties	Description
			Multiple days are represented as the sum of their numerical values. For example, Tuesday and Thursday = 4 + 16 = 20.
RecurrenceEndDateOnly	date	Create	The last date on which the holiday repeats. For multiday
		Filter	recurring events, this is the day on which the last occurrence starts.
		Nillable	
		Update	
RecurrenceInstance	picklist	Create	The frequency of the recurring holiday. For example, "2nd"
		Filter	or "3rd."
		Nillable	
		Restricted picklist	
		Update	
RecurrenceInterval	int	Create	The interval between recurring holidays.
		Filter	
		Nillable	
		Update	
RecurrenceMonthOfYear	picklist	Create	The month of the year on which the event repeats.
		Filter	
		Nillable	
		Restricted picklist	
		Update	
RecurrenceStartDate	date	Create	The date when the recurring holiday begins. Must be a date
		Filter	and time before RecurrenceEndDateOnly.
		Nillable	
		Update	
RecurrenceType	picklist	Create	Indicates how often the holiday repeats. For example, daily,
		Filter	weekly, or every Nth month (where "Nth" is defined in
		Nillable	Rediffencernotance,
		Restricted picklist	
		Update	
StartTimeInMinutes	int	Create	The start time of the holiday in minutes.
		Filter	

Field	Field Type	Field Properties	Description
		Nillable	
		Update	

Use this object to view and update holidays, which specify dates and times at which associated business hours and escalation rules are suspended. For more information on holidays, see "Setting Holidays" in the Salesforce online help.

Idea

Represents an idea on which users are allowed to comment and vote, for example, a suggestion for an enhancement to an existing product or process.

Corresponds to an SObject4 in which the ObjectType="Idea".

Supported Calls

Create(),Update(),Query(),Search(),Retrieve(),CreateObject(),GetDeleted(),GetUpdated(), DescribeSObjects()

Field	Field Type	Field Properties	Description
Body	textarea	Create Nillable Update	Description of the Idea.
Categories	multipicklist	Create Filter Nillable Update	Customizable multi-select picklist used to organize Ideas into logical groupings. Note: This field is only available if your organization has the Categories field enabled in Salesforce. This field is enabled by default in organizations created after API version 14 was released. If the Categories field is enabled, API versions 13 and earlier do not have access to either the Categories or Category fields.
Category	picklist	Create Filter Nillable Update	Customizable picklist of values used to organize Ideas into logical groupings. Note: This field is not available if your organization has the multi-select Categories field enabled.

Field	Field Type	Field Properties	Description
CommunityId	reference	Create Filter	The Community ID associated with the Idea. Once you create an Idea, you cannot change the Community ID associated with that Idea.
			Note: API version 12 does not support Community ID. If you create an idea in version 12, your idea is automatically posted to the oldest community that you have permission to access.
IsDeleted	boolean	Defaulted	Indicates whether the object has been moved to the Recycle
		Filter	Bin (true) or not (false). Deleting an idea also deletes its child objects IdeaComment and Vote.
IsHTML	boolean	Defaulted	Read-only. If this value is true, your organization has the Salesforce CRM Ideas HTML editor enabled and the Idea
		Filter	Body may contain HTML. If this value is false, the
		T HICI	HTML editor is disabled and the Idea Body only contains regular text. For information on enabling the HTML editor, search for "HTML Editor" in the Salesforce online help.
IsLocked	boolean	Filter	Read only. Indicates whether the idea has been locked because it was merged with a parent idea (true) or not (false). You cannot vote for or add comments to a locked idea.
LastCommentDate	dateTime	Filter	The date and time the last comment (child IdeaComment
		Nillable	object) was added.
LastCommentId	reference	Filter	Read only. The ID of the last comment (child IdeaComment
		Nillable	object).
NumComments	int	Filter	The number of comments (child IdeaComment objects) that
		Nillable	users have submitted for the given idea.
ParentIdeaID	reference	Nillable	The ID associated with this idea's parent idea. When multiple
		Filter	ideas are merged together, one idea becomes the parent (master) of the other ideas. The ParentIdeaID is automatically set when you merge ideas in Salesforce.
Status	picklist	Create	Customizable picklist of values used to specify the status of
		Filter	an idea.
		Nillable	
		Update	
Title	string	Create	The descriptive title of the idea.
		Filter	
		Update	

Field	Field Type	Field Properties	Description
VoteScore	double	Filter Nillable	The internal score of the Idea, used to sort Ideas on the Popular tab in the application user interface. The internal algorithm that determines the score gives older votes less weight than newer votes, simulating exponential decay. The score itself does not display in the application user interface.
VoteTotal	double	Filter Nillable	An Idea's total number of points. Each vote a user makes is worth ten points, therefore the value of this field is ten times the number of votes an idea has received.



Note: If you are importing Idea data into Salesforce and need to set the value for an audit field, such as CreatedDate, contact salesforce.com. Audit fields are automatically updated during API operations unless you request to set these fields yourself. For more information, see System Fields.

Usage

Use this object to track ideas, which are written suggestions on which users can vote and comment. For more information on ideas, see "About Ideas" in the Salesforce online help.

The boolean field IsPublished is not exposed in the public API for both entities. The value is defaulted at insert time based on whether the context user is a portal user or not. Soql queries also filter on this flag, so portal users can only see portal ideas and comments, and standard/admin users can only see internal ideas and comments.

IdeaComment

Represents a comment that a user has submitted in response to an idea.

Corresponds to an SObject4 in which the ObjectType="IdeaComment".

Supported Calls

```
Create(),Update(),Query(),Search(),Retrieve(),CreateObject(),GetDeleted(),GetUpdated(),
DescribeSObjects()
```

Field	Field Type	Field Properties	Description
CommentBody	textarea	Create	Body of the submitted comment.
		Filter	
		Nillable	
		Update	
IdeaId	reference	Create	ID of the idea on which this comment was made. For
		Filter	information on IDs, see ID Field Type.

Field	Field Type	Field Properties	Description
ISHTML	boolean	Defaulted on create Filter	Read-only. If this value is true, your organization has the Salesforce CRM Ideas HTML editor enabled, and the CommentBody field may contain HTML. If this value is false, the HTML editor is disabled and the CommentBody field only contains regular text. For information on enabling the HTML editor, search for "HTML Editor" in the Salesforce online help.



Note: If you are importing IdeaComment data into Salesforce and need to set the value for an audit field, such as CreatedDate, contact salesforce.com. Audit fields are automatically updated during API operations unless you request to set these fields yourself. For more information, see System Fields.

Usage

Use this object to track comments on ideas, which are users' text responses to ideas. For more information on ideas, see "About Ideas" in the Salesforce online help.

Lead

Represents a prospect or potential Opportunity.

Corresponds to an SObject4 in which the ObjectType="Lead".

Supported Calls

```
ConvertLead(),Create(),Update(),Query(),Search(),Retrieve(),CreateObject(),GetDeleted(),
GetUpdated(),DescribeSObjects()
```

Field	Field Type	Field Properties	Description
AnnualRevenue	currency	Create	Annual revenue for the company of the lead.
		Filter	
		Nillable	
		Update	
City	string	Create	City for the address of the lead.
		Filter	
		Nillable	
		Update	

Field	Field Type	Field Properties	Description
Company	string	Create	Required. Company of the lead.
		Filter	Note: In organizations where person account record
		Nillable	types have been enabled, if the value of the Company field is null the lead converts to a person account
		Update	See "Notes on Converting Leads" in the Salesforce online help for more information.
ConnectionReceivedID	reference	Filter	ID of the PartnerNetworkConnection that shared this record
		Nillable	with your organization. This field is only available if you have enabled Salesforce to Salesforce.
ConnectionSentID	reference	Filter	ID of the PartnerNetworkConnection that you shared this
		Nillable	Salesforce to Salesforce. Beginning with API version 15.0, the ConnectionSentID field is no longer supported. The ConnectionSentID field is still be visible, but the value is null. You can use the new PartnerNetworkRecordConnection object to forward records to connections.
ConvertedAccountId	reference	Filter	Object reference ID that points to the Account into which
Nillable the Lead has been conve ID Field Type.	the Lead has been converted. For information on IDs, see ID Field Type.		
ConvertedContactId	reference	Filter	Object reference ID that points to the Contact into whi
		Nillable	the Lead has been converted.
ConvertedDate	date	Filter	Date on which this Lead was converted.
		Nillable	
ConvertedOpportunityId	reference	Filter	Object reference ID that points to the Opportunity into
		Nillable	which the Lead has been converted.
Country	string	Create	Country for the address of the lead.
		Filter	
		Nillable	
		Update	
Description	textarea	Create	Description of the lead.
		Nillable	
		Update	
Email	email	Create	Email address for the lead.
		Filter	
		Nillable	
		Update	

Field	Field Type	Field Properties	Description
EmailBouncedDate	dateTime	Filter	If bounce management is activated and an email sent to the
		Nillable	lead bounced, the date and time the bounce occurred.
		Update	
EmailBouncedReason	string	Filter	If bounce management is activated and an email sent to the
		Nillable	lead bounced, the reason the bounce occurred.
		Update	
Fax	phone	Create	Fax number for the lead.
		Filter	
		Nillable	
		Update	
FirstName	string	Create	First name of the lead. Limited to 40 characters.
		Filter	
		Nillable	
		Update	
HasOptedOutOfEmail	boolean	Create	Indicates whether the lead has opted out of email (true) or
		Defaulted on create	not (false). Label 18 Email Opt Out.
		Filter	
		Update	
Industry	picklist	Create	Industry the lead works in.
		Filter	
		Nillable	
		Update	
IsConverted	boolean	Create	Indicates whether the Lead has been converted (true) or
		Defaulted on create	not (false). Label is Converted .
		Filter	
IsDeleted	boolean	Defaulted on create	Indicates whether the object has been moved to the Recycle Bin (true) or pot (false) Label is Deleted
		Filter	Dif (1100) of hot (12150). Laber is Deleted.
IsUnreadByOwner	boolean	Create	If true, lead has been assigned, but not yet viewed. See Unread
		Defaulted on create	Leads for more information. Label is Unread By Owner .
		Filter	

Field	Field Type	Field Properties	Description
		Update	
LastActivityDate	date	Filter Nillable	 Value is one of the following, whichever is the most recent: Due date of the most recent event logged against the record. Due date of the most recently closed task associated with the record.
LastName	string	Create Filter Nillable	Required. Last name of the lead. Limited to 80 characters.
LeadSource	picklist	Create Filter Nillable Update	Source from which the lead was obtained.
masterRecordId	reference	Filter Nillable	If this object was deleted as the result of a merge, this field contains the ID of the record that was kept. If this object was deleted for any other reason, or has not been deleted, the value is null.
MobilePhone	phone	Create Filter Nillable Update	Mobile phone number for the lead.
Name	string	Filter	Concatenation of FirstName and LastName. Limited to 121 characters.
NumberOfEmployees	int	Create Filter Nillable Update	Number of employees at the lead's company. Label is Employees .
OwnerId	reference	Create Defaulted on create Filter Update	ID of the owner of the lead.
Phone	phone	Create Filter	Phone number for the lead.

Field	Field Type	Field Properties	Description
		Nillable Update	
PostalCode	string	Create	Postal code for the address of the lead. Label is Zip/Postal
		Filter	Code.
		Nillable	
		Update	
Rating	picklist	Create	Rating of the lead.
		Filter	
		Nillable	
		Update	
Salutation	picklist	Create	Salutation for the lead.
		Filter	
		Nillable	
		Update	
State	string	Create	State for the address of the lead.
		Filter	
		Nillable	
		Update	
Status	picklist	Create	Status code for this converted lead. Status codes are defined
		Defaulted on create	in the lead status picklist and represented in the API by the LeadStatus object. See Lead Status Picklist for more information.
		Filter	
		Update	
Street	textarea	Create	Street number and name for the address of the lead.
		Filter	
		Nillable	
		Update	
Title	string	Create	Title for the lead, for example CFO or CEO.
		Filter	
		Nillable	
		Update	
Website	url	Create	Website for the lead.
		Filter	

Field	Field Type	Field Properties	Description
		Nillable	
		Update	



Note: If you are importing Lead data into Salesforce and need to set the value for an audit field, such as CreatedDate, contact salesforce.com. Audit fields are automatically updated during API operations unless you request to set these fields yourself. For more information, see System Fields.

Converted Leads

Leads have a special state to indicate that they have been converted into an Account, Contact, and optionally, an Opportunity. Your client application can convert leads via the ConvertLead() call. Users can alsoconvert leads through the Salesforce user interface. Once a lead has been converted, it is read-only. You cannot Update() or delete a converted lead. However, you can query converted Leads using the Query() call.

Leads have several fields that indicate their converted status. These special fields are read-only via the API. You cannot set these fields directly; they are set when converting the lead in the Salesforce user interface. The fields are:

- ConvertedAccountId
- ConvertedContactId
- ConvertedDate
- ConvertedOpportunityId
- IsConverted
- Status



Note: In organizations where person account record types have been enabled, if the value of the Company field is null, the lead converts to a person account. See "Notes on Converting Leads" in the Salesforce online help for more information.

Unread Leads

Leads have a special state to indicate that they have not been viewed or edited by the lead owner. In the Salesforce user interface, this is helpful for users to know which leads have been assigned to them but which they have not touched yet. The IsUnreadByOwner field is true if the lead owner has not yet viewed or edited the lead, and false if the lead owner has viewed or edited the lead at least once.

Lead Status Picklist

Each Status value corresponds to either a converted or unconverted status in the lead status picklist, as defined in the Salesforce user interface. To obtain the lead status values in the picklist, a client application can invoke the Query() call on the LeadStatus object.

You cannot convert a lead via the API by changing the Status field to one of the "converted" lead status values. When you convert qualified leads into an account, contact, and opportunity, you can select one of the "converted" status types to assign to the lead. Leads with a "converted" status type are no longer available in the Leads tab, although you can include them in reports.

Usage

To Update () a Lead or call ConvertLead (), your client application must be logged in with the "Edit" permission on leads.

When you Create(), Update(), Create() or Update() a lead, your client application can have the lead automatically assigned to one or more User objects based on assignment rules that have been configured in the Salesforce user interface.

To use this feature, your client application needs to set either of the following options (but not both) in the SOAP header (see SetSOAPHeader()) used in the Create() or Update() call:

Field	Field Type	Description
assignmentRuleId	reference	ID of the assignment rule to use. Can be an inactive assignment rule. If unspecified and useDefaultRule is true, then the default assignment rule is used.
useDefaultRule	boolean	Specifies whether to use the default rule for rule-based assignment (true) or not (false). The default rule is assigned by users in the Salesforce user interface.

LeadHistory

Represents the history of changes to the values in the fields of a lead.

Supported Calls

```
Query(),Retrieve(),GetDeleted(),GetUpdated(),DescribeSObjects(),GetDeleted(),
GetUpdated()
```

Fields

Field	Field Type	Field Properties	Description
Field	picklist	Filter Restricted picklist	The name of the field that was changed.
IsDeleted	boolean	Defaulted on create Filter	Indicates whether the object has been moved to the Recycle Bin (true) or not (false). Label is Deleted .
LeadId	reference	Filter	ID of the Lead. For information on IDs, see ID Field Type. Label is Lead ID.

Usage

Use this object to identify changes to a lead.

This object respects field level security on the parent object.

LeadOwnerSharingRule

Represents the rules for sharing a lead with users other than the owner.



Note: Contact salesforce.com customer support to enable access to this object for your organization.

Supported Calls

```
Create(),Update(),DescribeSObjects(),Query(),Retrieve(),CreateObject(),GetDeleted(),
GetUpdated()
```

Usage

Use these objects to manage the sharing rules for a particular lead. General sharing uses this object.

LeadShare

Represents a sharing entry on a Lead.

Corresponds to an SObject4 in which the ObjectType="LeadShare".

Supported Calls

Create(),Update(),DescribeSObjects(),Query(),Retrieve(),CreateObject()

Field	Field Type	Field Properties	Description
IsDeleted	boolean	Defaulted on create Filter	Indicates whether the object has been moved to the Recycle Bin (true) or not (false). Label is Deleted .
LeadAccessLevel	picklist	Filter Restricted picklist	 Level of access that the User or Group has to the Lead. The possible values are: Read Edit All (This value is not valid for Create() or Update() calls.) This field must be set to an access level that is higher than the organization's default access level for leads.
LeadId	reference	Filter	ID of the Lead associated with this sharing entry. This field cannot be updated. For information on IDs, see ID Field Type.

Field	Field Type	Field Properties	Description
RowCause	picklist	Filter Restricted picklist	 Reason that this sharing entry exists. Read-only. Values may include: Manual Sharing - The User or Group has access because a user with "All" access manually shared the Lead with them. Owner - The User is the owner of the Lead or is in a Role above the Case owner in the role hierarchy. ImplicitChild - User or Group has access to the Lead on the Account associated with this Lead. Sharing Rule - The User or Group has access via a Lead sharing rule.
UserOrGroupId	reference	Filter	ID of the User or Group that has been given access to the Lead. This field cannot be updated.

This object allows you to determine which users and groups can view and/or edit leads owned by other users. For more information, see Sharing.

If you attempt to create a record that matches an existing record, the Create() call updates any modified fields and returns the existing record.

LeadStatus

Represents the status of a Lead, such as Open, Qualified, or Converted.

Corresponds to an SObject4 in which the ObjectType="LeadStatus".

Supported Calls

```
Query(),Retrieve(),CreateObject(),DescribeSObjects()
```

Field	Field Type	Field Properties	Description
IsConverted	boolean	Defaulted on create Filter	Indicates whether this lead status value represents a converted lead (true) or not (false). Multiple lead status values can represent a converted lead. For more information, see ConvertLead().
IsDefault	boolean	Defaulted on create Filter	Indicates whether this is the default lead status value (true) or not (false) in the picklist.

Field	Field Type	Field Properties	Description
MasterLabel	string	Filter Nillable	Master label for this lead status value. This display value is the internal label that does not get translated.
SortOrder	int	Filter Nillable	Number used to sort this value in the lead status picklist. These numbers are not guaranteed to be sequential, as some previous lead status values might have been deleted.

This object represents a value in the lead status picklist (see Lead Status Picklist). The lead status picklist provides additional information about the status of a Lead, such as whether a given status value represents a converted Lead. Your client application can invoke the Query() call on the LeadStatus object to retrieve the set of values in the lead status picklist, and then use that information while processing Lead objects to determine more information about a given lead. For example, the application could test whether a given lead is converted based on its Status value and the value of the IsConverted property in the associated LeadStatus object.

This object is read-only via the API. With sufficient permissions, your client application can invoke the Query() and DescribeSObjects() calls on this object.

LeadTag

Associates a word or short phrase with a Lead.

Supported Calls

Create(),Query(),Retrieve(),DescribeSObjects()

Field	Field Type	Field Properties	Description
ItemId	reference	Create Filter	ID of the tagged item.
Name	string	Create Filter	Name of the tag. If this value does not already exist, a new TagDefinition is created and becomes the parent of this Tag object. Otherwise, a TagDefinition with the same name becomes the parent of this Tag object. Parent relationships are created automatically.
TagDefinitionId	reference	Filter	ID of the parent TagDefinition object that owns the tag.
Туре	picklist	Create Filter Restricted picklist	 Defines the visibility of a tag. Possible value are: Public: The tag can be viewed and manipulated by all users in an organization Personal: The tag can be viewed or manipulated only by a user with a matching OwnerId

LeadTag stores the relationship between its parent TagDefinition and the Lead being tagged. Tag objects act as metadata, allowing users to describe and organize their data.

When a tag is deleted, its parent TagDefinition will also be deleted if the name is not being used; otherwise, the parent remains. Deleting a TagDefinition sends it to the recycle bin, along with any associated tag entries.

For more information on tags, see "About Tagging" in the Salesforce online help.

LineitemOverride

A forecast override of a line item on an Opportunity.

Supported Calls

Query(),Retrieve(),GetUpdated(),DescribeSObjects()

Special Access Rules

This object is only accessible if your organization has enabled the customizable forecasting feature, which can be done using the Salesforce user interface. Requires the "View All Data" permission.

Field	Field Type	Field Properties	Description
AmountInherited	boolean	Defaulted on create Filter	Indicates whether the overridden amount rolls up through the forecast hierarchy (true), or was overridden by the owner of this record (false).
ForecastCategoryInherited	boolean	Defaulted on create Filter	Indicates whether the overridden forecast category rolls up through the forecast hierarchy (true), or was overridden by the owner of this record (false).
IsDeleted	boolean	Defaulted on create Filter	Indicates whether the object has been moved to the Recycle Bin (true) or not (false), usually because the parent Opportunity is moved to the Recycle Bin.
OpportunityId	reference	Filter	ID of the associated Opportunity. For information on IDs, see ID Field Type.
OpportunityLineItemId	reference	Filter	ID of the associated OpportunityLineItem.
OverrideAmount	currency	Filter	The total monetary amount of the line item, which may be overridden.
OverrideForecastCategory	picklist	Filter Restricted picklist	The forecast category of the line item. Can be overridden.
OverrideQuantity	double	Filter	The quantity of the line item, which may be overridden.

Field	Field Type	Field Properties	Description
OverrideUnitPrice	currency	Filter	The monetary amount of the unit price. Can be overridden.
OwnerId	reference	Filter	ID of the owner of this record.
QuantityInherited	boolean	Defaulted on create Filter	Indicates whether the overridden quantity rolls up through the forecast hierarchy (true), or is overridden by the owner of this record (false).
UnitPriceInherited	boolean	Defaulted on create Filter	Indicates whether the overridden unit price rolls up through the forecast hierarchy (true), or was overridden by the owner of this record (false).

This read-only object for customizable forecasting has a child-parent relationship with OpportunityOverride.

MailMergeTemplate

Represents a mail merge template (a Microsoft Word document) used for performing mail merges for your organization.

Corresponds to an SObject4 in which the ObjectType="MailMergeTemplate".

Supported Calls

```
Create(),Update(),DescribeSObjects(),Query(),Retrieve(),CreateObject(),GetDeleted(),
GetUpdated()
```

Special Access Rules

- All users can view this object, but you need the "Customize Application" permission to modify it.
- Customer Portal users cannot access this object.

Field	Field Type	Field Properties	Description
Body	base64	Create	Required. Microsoft Word document to use as a mail merge template. Due to limitations with Microsoft Word mail merge templates, your client application can specify the Body field in the Create() call but not in the Update() call. Limit: 5 MB.
BodyLength	int	Filter Nillable	Length of the Microsoft Word document.
Description	string	Create Filter	Required. Text description of this mail merge template. Limit: 255 characters.

Field	Field Type	Field Properties	Description
		Update	
Filename	string	Create Filter	Required. File name of the Microsoft Word document that was uploaded as a mail merge template. Limit: 255 characters in length.
IsDeleted	boolean	Defaulted on create Filter	Indicates whether the record has been moved to the Recycle Bin (true) or not (false). Label is Deleted .
LastUsedDate	dateTime	Filter Nillable	Date and time when this MailMergeTemplate was last used.
Name	string	Create Filter Update	Required. Name of this mail merge template.

Use this object to manage mail merge templates for your organization.

Name

This object is used to retrieve information from related records where the related record may be from more than one object type (a polymorphic foreign key). For example, the owner of a case can be either a user or a group (queue). This object allows retrieval of the owner name, whether the owner is a user or a group (queue). You can use a describe call to access the information about parents for an object, or you can use the who, what, or owner fields (depending on the object) in SOQL queries. This object cannot be directly accessed.

Supported Calls and Queries

 $Calls: {\tt DescribeSObjects()}$

SOQL: SELECT (including the WHERE clause), ORDER BY

Field	Field Type	Field Properties	Description
Alias	string	Filter Nillable	The user alias. This field contains a value only if the related record is a user.
Email	string	Filter Nillable	The email address of the user. This field contains a value only if the related record is a user .

Field	Field Type	Field Properties	Description
FirstName	string	Filter	If the related record is a user, contact, or lead, the first name.
		Nillable	
IsActive	boolean	Filter	Indicates whether the related record is an active user (true)
		Nillable	or not $(false)$. This field contains a value only if the related record is a user.
LastName	string	Filter	If the related record is a user, contact, or lead, the last name.
		Nillable	
Name	string	Filter	Name of the parent of the object queried. If the parent is a
		Nillable	user, contact, or lead, the value is a concatenation of firstname and lastname fields of the related record.
Phone	string	Filter	The phone number of the user. This field contains a value
		Nillable	only if the related record is a user.
Profile	reference	Filter	The Profile of the user. Only populated if the related record
		Nillable	is a user.
ProfileId	reference	Filter	ID of the user's Profile. For information on IDs, see ID Field
	Nillable	Type. Only populated if the related record.	
Title	string	Filter	The title of the user, for example CFO or CEO.
		Nillable	
Туре	string	Filter	A list of the types of SObject4s that can be an owner of this $1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 $
		Nillable	example, return only the leads owned by a user.
		Restricted picklist	
Username	string	Filter	Contains the name that a user enters to log into the API or
		Nillable	form of an email address, and is only populated if the related record is a user.
UserRole	picklist	Filter	Name of the $Role$ played by the user. Only populated for
		Nillable	user rows.
UserRoleId	reference	Filter	The ID of the user role associated with this object.
		Nillable	

To query on relationships where the parent may be more than one type of object, use the who, what, and owner fields. For more information about these types of queries and the who, owner, and what fields, see Understanding Polymorphic Keys and Relationships.

Note

Represents a note, which is text associated with a custom object or a standard object, such as a Contact, Contract, or Opportunity. Corresponds to an SObject4 in which the ObjectType="Note".

Supported Calls

Create(),Update(),DescribeSObjects(),Query(),Search(),Retrieve(),CreateObject(), GetDeleted(),GetUpdated()

Field	Field Type	Field Properties	Description
Body	textarea	Create	Body of the note. Limited to 32 KB.
		Nillable	
		Update	
IsDeleted	boolean	Defaulted on create	Indicates whether the object has been moved to the Recycle Bin (true) or not (false). Label is Deleted .
		Filter	
IsPrivate	boolean	Create	If true, only the note owner or a user with the "Modify All
		Defaulted on create	Data" permission can view the note or query it via the API. Note that if a user who does not have the "Modify All Data" permission sets this field to true on a note that they do not
		Filter	own, then they can no longer Query() or Update() the
		Update	note. Label is Private .
OwnerId	reference	Create	ID of the user who owns the note.
		Defaulted on create	
		Filter	
		Update	
ParentId	reference	Create	Required. ID of the object associated with the note.
		Filter	
Title	string	Create	Title of the note.
		Filter	
		Nillable	
		Update	

Use this object to manage notes for an object.

NoteAndAttachment

This read-only object contains all notes and attachments associated with an object.

Supported Calls

DescribeGlobal(),DescribeSObject(),DescribeSObjects()

Fields

Field	Field Type	Field Properties	Description
IsDeleted	boolean	Defaulted on create Filter	Indicates whether the object has been moved to the Recycle Bin (true) or not (false). Label is Deleted .
IsNote	boolean	Defaulted on create Filter	Indicates whether the object contains a note (true) or an attachment (false).
IsPrivate	boolean	Defaulted on create Filter	If true, only the note owner or a user with the "Modify All Data" permission can view the note or query it via the API. Note that if a regular user who does not have "Modify All Data" permission sets this field to true on a note that they do not own, then they can no longer Query() or Update() that note. Label is Private .
OwnerId	reference	Filter	ID of the user who owns the note and attachment.
ParentId	reference	Filter	ID of the parent object.
Title	string	Filter Nillable	Title of the note.

Usage

Use this object to list all notes and attachments for an object.

To retrieve this object, issue a describe call on an object, which will return a query result for each activity since the object was created. You cannot use Query().

NoteTag

Associates a word or short phrase with a Note.

Supported Calls

Create(),Query(),Retrieve(),DescribeSObjects()

Fields

Field	Field Type	Field Properties	Description
ItemId	reference	Create Filter	ID of the tagged item.
Name	string	Create Filter	Name of the tag. If this value does not already exist, a new TagDefinition is created and becomes the parent of this Tag object. Otherwise, a TagDefinition with the same name becomes the parent of this Tag object. Parent relationships are created automatically.
TagDefinitionId	reference	Filter	ID of the parent TagDefinition object that owns the tag.
Туре	picklist	Create Filter Restricted picklist	 Defines the visibility of a tag. Possible value are: Public: The tag can be viewed and manipulated by all users in an organization Personal: The tag can be viewed or manipulated only by a user with a matching OwnerId

Usage

NoteTag stores the relationship between its parent TagDefinition and the Note being tagged. Tag objects act as metadata, allowing users to describe and organize their data.

When a tag is deleted, its parent TagDefinition will also be deleted if the name is not being used; otherwise, the parent remains. Deleting a TagDefinition sends it to the recycle bin, along with any associated tag entries.

For more information on tags, see "About Tagging" in the Salesforce online help.

OpenActivity

This read-only object contains all open tasks and events associated with an object. OpenActivity fields related to calls are only available with Salesforce CRM Call Center.

Supported Calls

```
DescribeGlobal(),DescribeSObject(),DescribeSObjects()
```

Field	Field Type	Field Properties	Description
ActivityDate	date	Filter	The due date of the task, or for an activity, then the due date if IsAllDayEvent is set to true.
		Nillable	This field has a timestamp that is always set to midnight in the Coordinated Universal Time (UTC) time zone. The timestamp is not relevant; do not attempt to alter it in order to accommodate time zone differences. Label is Date .
ActivityType	picklist	Filter Nillable	One of the following values: Call, Meeting, or Other. Label is Type .
CallDisposition	string	Filter Nillable	Represents the result of a given call, for example, "we'll call back," or "call unsuccessful." Limit is 255 characters.
CallDurationInSeconds	int	Filter Nillable	Duration of the call in seconds.
CallObject	string	Filter Nillable	Name of a call center. Limit is 255 characters.
CallType	picklist	Filter Nillable	The type of call being answered: Inbound, Internal, or Outbound.
Description	textarea	Nillable	Description of the task or event. Limit is 32 KB.
DurationInMinutes	int	Filter Nillable	Length of the event or task.
IsAllDayEvent	boolean	Defaulted on create Filter	If true, the activity is an event and the ActivityDate is used to define the date of the event. If false, the activity may be a task or an event. Label is All Day Event .
IsClosed	boolean	Defaulted on create Filter	For tasks only, indicates whether the task was completed (true) or not (false). This field is set indirectly by setting the Status fieldeach picklist value has a corresponding IsClosed value. Label is Closed .
IsDeleted	boolean	Defaulted on create Filter	Indicates whether the record has been moved to the Recycle Bin (true) or not (false). Label is Deleted .
IsTask	boolean	Defaulted on create Filter	If true, the activity is a task, if false, it is an event. Label is Task .
		1 11001	

Field	Field Type	Field Properties	Description
IsVisibleInSelfService	boolean	Defaulted on create	If true, the activity can be viewed in the self-service portal. Label is Visible in Self-Service .
		Filter	
Location	string	Filter	If an event, the location of the event. If not, the value is null.
		Nillable	
OwnerId	reference	Filter	ID of the user who owns the task or event.
		Nillable	
Priority	picklist	Filter	If a task, the importance of the task, such as high, normal,
		Nillable	or low.
Status	picklist	Filter	For tasks, the current status of the task, for example In
		Nillable	value for IsClosed. To obtain picklist values, you can Query () on the TaskStatus object.
Subject	comobobox	Filter	Subject line of the task or event.
		Nillable	
WhatId	reference	Filter	ID of the related object (Account, Campaign, Case,
		Nillable	Opportunity/Account ID.
WhoId	reference	Filter Nillable	ID of the related Contact or Lead. If Whold refers to a lead, then the Whatld field must be empty. Label is Contact/Lead ID.

This object helps you replicate the related list functionality of the Salesforce user interface. To use this object, use the following procedure:

- 1. Optionally, issue a describe call against the object whose activities you wish to query, to get a suggestion of the correct SOQL to use.
- 2. Issue a SOQL relationship query with a main clause that references the object, and an inner clause that references the activity history, for example:

```
SELECT (SELECT ActivityDate, Description from ActivityHistories) FROM Account WHERE Name Like 'XYZ%'
```

or

SELECT (SELECT ActivityDate, Description from OpenActivities) FROM Account WHERE Name Like 'XYZ%'

The Salesforce user interface enforces sharing rules, filtering out related list items that a user does not have permission to see.

In order to prevent performance issues while still providing the related list functionality, there are some restrictions on users who do not have "View All Data" permission. Such users must comply with the following restrictions:

- In the main clause of the relationship query, you can reference only one record. For example, you cannot filter on all records where the account name starts with A, but must reference a single account record.
- You cannot use WHERE clauses.
- You must specify a limit to the number of rows returned, less than 500.
- You must sort on ActivityDate and LastModifiedDate, descending order: ORDER BY ActivityDate DESC, LastModifiedDate DESC

You cannot use Query () directly on this object.

Opportunity

Represents an opportunity, which is a sale or pending deal.

Corresponds to an SObject4 in which the ObjectType="Opportunity".

Supported Calls

Create(),Update(),Query(),Search(),Retrieve(),CreateObject(),GetDeleted(),GetUpdated(), DescribeSObjects()

Field	Field Type	Field Properties	Description
AccountId	reference	Create	Id of the account associated with this opportunity.
		Filter	
		Nillable	
		Update	
Amount	currency	Create	Estimated total sale amount. For opportunities with products,
		Filter	the amount is the sum of the related products. Any attempt to update this field, if the record has products, will be ignored.
		Nillable	The update call will not be rejected, and other fields will be
		Update	updated as specified, but the Amount will be unchanged.
CampaignId	reference	Create	ID of a related Campaign. This field is defined only for those
		Filter	organizations that have Campaigns enabled as a feature. The User must have read access rights to the cross-referenced
		Nillable	Campaign object in order to Create() or Update() that
		Update	campaign into this field on the opportunity.
CloseDate	date	Create	Required. The date when the opportunity is expected to close.
		Filter	
Field	Field Type	Field Properties	Description
----------------------	------------	----------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
		Update	
ConnectionReceivedID	reference	Filter Nillable	ID of the PartnerNetworkConnection that shared this record with your organization. This field is only available if you have enabled Salesforce to Salesforce.
ConnectionSentID	reference	Filter Nillable	ID of the PartnerNetworkConnection that you shared this record with. This field is only available if you have enabled Salesforce to Salesforce. Beginning with API version 15.0, the ConnectionSentID field is no longer supported. The ConnectionSentID field is still be visible, but the value is null. You can use the new PartnerNetworkRecordConnection object to forward records to connections.
CurrencyIsoCode	picklist	Create Defaulted on create	Available only for organizations with the multicurrency feature enabled. Contains the ISO code for any currency allowed by the organization.
		Filter Nillable	If the organization has multicurrency and a Pricebook2 is specified on the opportunity (i.e., the Pricebook2Id field is not blank) then the currency value of this field must match
		Restricted picklist	the currency of the PricebookEntry objects that are associated with any opportunity line items it has.
		Update	
Description	textarea	Create	Text description of the opportunity. Limit: 32,000 characters.
		Nillable	
FynactadRavanua	currency	Filter	A read-only field that is equal to the product of the
	currency	Nillable	opportunity Amount field and the Probability. You cannot directly set this field, but you can indirectly set it by setting the Amount or Probability fields.
Fiscal	string	Filter	If fiscal years are not enabled, the name of the fiscal quarter or period in which the opportunity CloseDate falls. Value should be in YYY Q format, for example, '2006 1' for first quarter of 2006.
FiscalQuarter	int	Filter	Represents the fiscal quarter. Valid values are 1, 2, 3, or 4.
		Nillable	
FiscalYear	int	Filter Nillable	Represents the fiscal year, for example, 2006.
ForecastCategory	picklist	Create	A restricted picklist field. It is implied, but not directly
		Defaulted on create	controlled, by the StageName field. You can override this field to a different value than is implied by the StageName value. The values of this field are fixed enumerated values
		Filter and	The field labels are localized to the language of the user

Field	Field Type	Field Properties	Description
		Nillable Restricted	performing the operation, if localized versions of those labels are available for that language in the Salesforce user interface.
		picklist Update	In API version 12.0 and later, the value of this field is automatically set based on the value of the ForecastCategoryName and cannot be updated any other way. The field properties Create, Defaulted on create, Nillable, and Update are not available in version 12.0.
ForecastCategoryName	picklist	Create	Available in API version 12.0 and later.
		Defaulted on create	The name of the forecast category. It is implied, but not directly controlled by the StageName field You can override
		Filter	this field to a different value than is implied by the
		Nillable	StageName value.
		Update	
HasOpportunityLineItem	boolean	Defaulted on create Filter	Read-only field that indicates whether the opportunity has associated line items. A value of true means that Opportunity line items have been created for the opportunity. An opportunity can have opportunity line items only if the opportunity has a price book. The opportunity line items must correspond to PricebookEntry objects that are listed in the opportunity Pricebook2. However, you can insert opportunity line items on an opportunity that does not have an associated Pricebook2. For the first opportunity line item that you insert on an opportunity without a Pricebook2, the API automatically sets the Pricebook2Id field, if the opportunity line item corresponds to a PricebookEntry in an active Pricebook2 that has a CurrencyIsoCode field that matches the CurrencyIsoCode field of the opportunity. If the Pricebook2 is not active or the CurrencyIsoCode fields do not match, then the API returns an error. You cannot Update() the Pricebook2Id or PricebookId fields if opportunity line items exist on the Opportunity. You must delete the line items before attempting to update the PricebookId field.
IsClosed	boolean	Defaulted on create Filter	Directly controlled by StageName. You can query and filter on this field, but you cannot directly set it in a Create(), or Update() request. It can only be set via StageName. Label is Closed
IsDeleted	boolean	Defaulted on create Filter	Indicates whether the record has been moved to the Recycle Bin (true) or not (false). Label is Deleted .
IsWon	boolean	Defaulted on create Filter	Directly controlled by StageName. You can query and filter on this field, but you cannot directly set it in a Create(), or Update() request. It can only be set via StageName. Label is Won .

Field	Field Type	Field Properties	Description
LastActivityDate	date	Filter	Value is one of the following, whichever is the most recent:
		Nillable	• Due date of the most recent event logged against the record.
			• Due date of the most recently closed task associated with the record.
LeadSource	picklist	Create	The source of this opportunity, such as Advertisement or
		Filter	Irade Snow.
		Nillable	
		Update	
Name	string	Create	Required. A name for this opportunity. Limit: 80 characters.
		Filter	
		Update	
NextStep	string	Create	Description of next task in closing opportunity. Limit: 255
		Filter	characters.
		Nillable	
		Update	
OwnerId	reference	Create	ID of the User who has been assigned to work this
		Defaulted	opportunity.
		Filter	this field has different consequences depending on your version of the API:
		Update	• For API version 12.0 and later, sharing records are kept, as they are for all objects.
			• For API version before 12.0, sharing records are deleted.
Pricebook2Id	reference	Create	ID of a related Pricebook2 object. The Pricebook21d field
		Defaulted	indicates which Pricebook2 applies to this opportunity. The
		on create	that have products enabled as a feature. You can specify values
		Filter	for only one field (Pricebook2Id or PricebookId)—not both fields. For this reason, both fields are declared nillable
		Nillable	both fields. For this reason, both fields are declared fillable.
		Update	
PricebookId	reference	Create	Deprecated as of version 3.0. As of version 8.0, the Pricebook
		Defaulted on create	instead, specifying the ID of the Pricebook2 object.
		Filter	
		Nillable	

Field	Field Type	Field Properties	Description
		Update	
Probability	percent	Create Defaulted on create Filter Nillable Update	Percentage of estimated confidence in closing the opportunity. It is implied, but not directly controlled, by the StageName field. You can override this field to a different value than what is implied by the StageName.
StageName	picklist	Create Filter Update	Required. Current stage of this record. The StageName field controls several other fields on an opportunity. Each of the fields can be directly set or implied by changing the StageName field. In addition, the StageName field is a picklist, so it has additional members in the returned Field4 object to indicate how it affects the other fields. To obtain the stage name values in the picklist, invoke the Query() call on the OpportunityStage object. If the StageName is updated, then the ForecastCategoryName, IsClosed, IsWon, and Probability are automatically updated based on the stage-category mapping.
TotalOpportunityQuantity	double	Create Filter Nillable Update	The number of items included in this opportunity. Used in quantity-based forecasting.
Туре	picklist	Create Filter Nillable Update	The type of opportunity. For example, Existing Business or New Business. Label is Opportunity Type .

Note: If you are importing Opportunity data into Salesforce and need to set the value for an audit field, such as CreatedDate, contact salesforce.com. Audit fields are automatically updated during API operations unless you request to set these fields yourself. For more information, see System Fields.

Usage

Use the Opportunity object to manage information about a sale or pending deal. To Update() an Opportunity, your client application needs "Edit" permission on opportunities. Client applications can Create(), Update(), and Query() Attachments associated with an opportunity via the API. For a visual diagram of the relationships between Opportunity and other objects, see Product and Schedule Objects.



Note: With Spring '09, workflow rules, validation rules and Apex triggers on opportunities and opportunity products will fire when an update to a child opportunity product or schedule causes an update to the parent record. This means your custom application logic will be enforced when there are updates to the parent record, ensuring higher data quality and compliance with your organization's business policies.

By default, all organizations signed up after the Spring '09 release will see this behavior. Starting in March 09, salesforce.com will begin rolling out the New Opportunity Save Behavior update in phases. Once available, you will see the update on the Critical Updates page. The update will be auto-enabled for organizations not currently using any of the customizations listed below. These customers will be required to review the update and take the necessary actions in order to enable the update prior to its auto-activation date. This change is effective in all versions of the API once the New Opportunity Save Behavior update is activated for an organization.

For more information, see What is the Spring '09 New Opportunity Save Behavior Update? in the Salesforce online help.

OpportunityCompetitor

Represents a competitor on an Opportunity.

Corresponds to an SObject4 in which the ObjectType="OpportunityCompetitor".

Supported Calls

```
Create(),Update(),DescribeSObjects(),Query(),Retrieve(),CreateObject(),GetDeleted(),
GetUpdated()
```

Field	Field Type	Field Properties	Description
CompetitorName	comobobox	Create	Name of the competitor.
		Filter	
		Nillable	
		Update	
IsDeleted bo	boolean	Defaulted on create	Indicates whether the record has been moved to the Recycle Bin (true) or not (false) Label is Deleted
		Filter	Diff (e1 de) of hot (1419e). Eaber is Deleted .
OpportunityID ref	reference	Create	Required. ID of the associated Opportunity. For information
		Filter	on IDs, see ID Field Type.
Strengths	string	Create	Description of the competitor's strengths. Limit: 1,000
		Filter	characters.
		Nillable	
		Update	
Weaknesses	string	Create	Description of the competitor's weaknesses. Limit: 1,000
		Filter	characters.
		Nillable	
		Update	

Use this object to manage competitors on an Opportunity, associating multiple competitors on a opportunity and specifying the strengths and weaknesses of each competitor.

OpportunityContactRole

Represents the role that a Contact plays on an Opportunity.

Corresponds to an SObject4 in which the ObjectType="OpportunityContactRole".

Supported Calls

Create(),Update(),DescribeSObjects(),Query(),Retrieve(),CreateObject(),GetDeleted(), GetUpdated()

Field	Field Type	Field Properties	Description
ContactId	reference	Create Filter Update	ID of an associated Contact. The API applies user access rights to the associated Opportunity for this object, but not to the associated Contact. The API may return rows from a query on this object that include this field's values for contacts to which the user does not have sufficient access rights. It may also return values for this field for contacts that have been deleted. In either case, the client must perform a query on the contact table for this field's value to determine whether the Contact is accessible to the user and has not been deleted.
IsDeleted	boolean	Defaulted on create Filter	Indicates whether the record has been moved to the Recycle Bin (true) or not (false). Label is Deleted .
IsPrimary	boolean	Create Defaulted on create Filter Update	Indicates whether the associated Contact plays the primary role on the Opportunity (true) or not (false). Each Opportunity has only one primary contact. Label is Primary .
OpportunityId	reference	Create Filter	Required. ID of an associated Opportunity. This field is non-nullable, and it cannot be updated. You must provide a value for this field on Create(). You cannot change it after it has been created.
Role	picklist	Create Filter Nillable Update	Name of the role played by the associated Contact on the Opportunity, such as Business User or Decision Maker.

Records of this object type appear in the Salesforce user interface on the Opportunity detail page. Like most other objects, records of this object type have their own unique ID that you use when updating or deleting records.

Although allowed, we do not recommend that you create multiple relationships between the same Opportunity and a Contact.

OpportunityFieldHistory

Represents the history of changes to the values in the fields of an opportunity. This object is available in versions 13.0 and later.

Supported Calls

Query(),Retrieve(),GetDeleted(),GetUpdated(),and DescribeSObjects()

Fields

Field	Field Type	Field Properties	Description
Field	picklist	Filter Restricted	The name of the field that was changed.
		picklist	
IsDeleted	boolean	Defaulted on create Filter	Indicates whether the object has been moved to the Recycle Bin (true) or not (false). Label is Deleted .
OpportunityId	reference	Filter	ID of the Opportunity. For information on IDs, see ID Field Type. Label is Opportunity ID .

Usage

Use this object to identify changes to any fields on an Opportunity. The OpportunityHistory object represents the history of a change to the Amount, Probability, Stage, or Close Date fields of an Opportunity.

This object respects field level security on the parent object.

OpportunityHistory

Represents the stage history of an Opportunity.

Corresponds to an SObject4 in which the ObjectType="OpportunityHistory".

Supported Calls

Query(),Retrieve(),CreateObject(),GetDeleted(),GetUpdated(),DescribeSObjects()

Fields

Field	Field Type	Field Properties	Description
Amount	double	Filter	Estimated total sale amount.
		Nillable	
CloseDate	date	Filter	Date when the opportunity is expected to close.
		Nillable	
ExpectedRevenue	currency	Filter	Calculated revenue based on the Amount and Probability
		Nillable	fields.
ForecastCategory	picklist	Filter	Category that determines the column in which an opportuni is totaled in a forecast. Label is To ForecastCategory .
		Nillable	
		Restricted picklist	
IsDeleted	boolean	Defaulted on create	Indicates whether the record has been moved to the Recycle Bin (true) or not (false).
		Filter	Label is Deleted .
OpportunityId	reference	Filter	ID of the associated Opportunity.
Probability	percent	Filter	Percentage of estimated confidence in closing the opportunity.
		Nillable	
StageName	picklist	Filter	Name of the current stage of the opportunity (for example, Prospect or Proposal).

Usage

This object represents the history of a change to the Amount, Probability, Stage, or Close Date fields of an Opportunity. The OpportunityFieldHistory object represents the history of a change to any of the fields of an Opportunity. To obtain information about how a particular opportunity is progressing, Query() all of the OpportunityHistory objects associated with a given Opportunity. Please note that if an opportunity's Amount, Probability, Stage, or Close Date fields have not changed, nothing will be returned in the OpportunityHistory objects. In this case, Query() all of the OpportunityFieldHistory objects associated with a given Opportunity to get more information about changes to the opportunity.

This object is read-only. The system generates a new record whenever a user or client application changes the value of any of the above fields; the then-current values of all of these major fields are saved in the newly-generated object.

This object respects field-level security on the parent object.



Note: The record is automatically deleted if its parent Opportunity is deleted.

OpportunityLineItem

Represents an opportunity line item, which is a member of the list of Product2 products associated with an Opportunity, along with other information about those products on that opportunity.

Corresponds to an SObject4 in which the ObjectType="OpportunityLineItem".

Supported Calls

Create(),Update(),DescribeSObjects(),Query(),Retrieve(),CreateObject(),GetDeleted(), GetUpdated()

Special Access Rules

The user must have the "Edit" permissions on Opportunity records in order to Create() or Update() opportunity line items on an opportunity.

Field	Field Type	Field Properties	Description
ConnectionReceivedID	reference	Filter Nillable	ID of the PartnerNetworkConnection that shared this record with your organization. This field is only available if you have enabled Salesforce to Salesforce.
ConnectionSentID	reference	Filter Nillable	ID of the PartnerNetworkConnection that you shared this record with. This field is only available if you have enabled Salesforce to Salesforce. Beginning with API version 15.0, the ConnectionSentID field is no longer supported. The ConnectionSentID field is still be visible, but the value is null. You can use the new PartnerNetworkRecordConnection object to forward records to connections.
CurrencyIsoCode	picklist	Filter Restricted picklist	Available only for organizations with the multicurrency feature enabled. Contains the ISO code for any currency allowed by the organization. If the organization has multicurrency enabled, and a Pricebook2 is specified on the parent opportunity (i.e., the Pricebook21d field is not blank on the opportunity referenced by this object's OpportunityId), then the value of this field must match the currency of the CurrencyIsoCode field on the PricebookEntry objects that are associated with this object.
Description	string	Create Filter Nillable Update	Text description of the opportunity line item. Limit: 255 characters.

Field	Field Type	Field Properties	Description
HasQuantitySchedule	boolean	Defaulted on create	Read-only. Indicates whether a quantity schedule has been created for this object (true) or not (false).
		Filter	
HasRevenueSchedule	boolean	Defaulted on create	Read-only. Indicates whether a revenue schedule has been created for this object (true) or not (false).
		Filter	If this object has a revenue schedule, the Quantity and TotalPrice fields cannot be updated. In addition, the Quantity field cannot be updated if this object has a quantity schedule. The API ignores any attempt to update these fields. The Update() call will not be rejected but the updated values will be ignored.
IsDeleted	boolean	Defaulted on create Filter	Indicates whether the record has been moved to the Recycle Bin (true) or not (false). Label is Deleted . This flag is only set to true when the parent opportunity is deleted, and is set to false again if the opportunity is undeleted.
			Deleting an OpportunityLineItem directly via the API completely removes it, and it cannot be undeleted.
ListPrice	currency	Filter	Corresponds to the UnitPrice on the PricebookEntry that
		Nillable	is associated with this line item, which can be in the standard pricebook or a custom pricebook. A client application can use this information to show whether the unit price (or sales price) of the line item differs from the pricebook entry list price.
OpportunityId	reference	Create	Required. ID of the associated Opportunity.
		Filter	
PriceBookEntryId	reference	Create	Required. ID of the associated PricebookEntry. Exists of for those organizations that have Products enabled as a
		Filter	feature. You can specify values for either this field or
		Nillable	ProductId, but not both. For this reason, both fields are declared nillable.
ProductId	reference	Create	ID of the associated Product object. This field has been
		Filter	compatibility. The Product object is unavailable beginning
	Nillable with version 8.0. Use the Prices	with version 8.0. Use the PriceBookEntryId field instead, specifying the ID of the PricebookEntry object.	
Quantity	double	Create	Read-only if this record has a quantity schedule, a revenue
		Filter	schedule, or both a quantity and a revenue schedule.
		Update	
ServiceDate	date	Create	Date when the product revenue will be recognized and the
Filte	Filter	product quantity will be shipped. Whether this value is used by customizable forecasting depends upon the Forecast Date setting for the organization:	

Field	Field Type	Field Properties	Description
		Nillable Update	 Opportunity Close Date—ServiceDate is ignored. Product Date—ServiceDate is used if not null. Schedule Date—ServiceDate is used if not null and there are no revenue schedules present for this line item, that is, there are no OpportunityLineItemSchedule objects with a field Type value of Revenue that are children of this object.
SortOrder	int	Filter Nillable	Number indicating the sort order selected by the user. Client applications can use this to match the sort order in Salesforce.
TotalPrice	currency	Create Defaulted on create Filter Nillable Update	This field is deprecated but exists in order to provide backward compatibility. It represents the total price of the OpportunityLineItem. If you do not specify UnitPrice, this field is required. In a given Update () call, you can change either this value or the UnitPrice, but not both at the same time. This field is nillable, but you cannot set both TotalPrice and UnitPrice to null in the same Update () call. To insert the TotalPrice for an opportunity line item via the API (given only a unit price and the quantity), calculate this field as the unit price multiplied by the quantity. This field is read-only if the opportunity line item has a revenue schedule. If the opportunity line item does not have a schedule or only has quantity schedule, this field can be updated.
UnitPrice	currency	Create Defaulted on create Filter Nillable Update	This field or TotalPrice is required, you cannot specify both. The unit price for this opportunity line item. In the Salesforce user interface, this field's value is calculated by dividing the total price of the opportunity line item by the quantity listed for that line item. Label is Sales Price .

An Opportunity can have OpportunityLineItems only if the Opportunity has a Pricebook2. An OpportunityLineItem must correspond to a Product2 that is listed in the opportunity's Pricebook2. For information about inserting OpportunityLineItem for an opportunity that does not have an associated Pricebook2 or any existing line items, see Effects on Opportunities.

This object is defined only for those organizations that have products enabled as a feature. If the organization does not have the products feature, this object does note appear in the DescribeGlobal() call, and you cannot use DescribeSObjects() or Query() with the OpportunityLineItem object.

For a visual diagram of the relationships between OpportunityLineItem and other objects, see Product and Schedule Objects.



Note: If multicurrency option has been enabled, the CurrencyIsoCode field will be present. It cannot be modified, and is always set to the value of the CurrencyIsoCode of the parent Opportunity. For more information about CurrencyIsoCode fields, see Currency Field Type.

Effects on Opportunities

Opportunities that have associated OpportunityLineItems are affected in the following ways:

- Creating an OpportunityLineItem increments the Opportunity Amount value by the TotalPrice of the OpportunityLineItem. Additionally, inserting an OpportunityLineItem increments the ExpectedRevenue on the opportunity by the TotalPrice times the opportunity Probability.
- The Opportunity Amount becomes a read-only field when the opportunity has line items. The API ignores any attempt to update this field on an opportunity with line items. The Update () call will not be rejected, but the updated value will be ignored.
- You cannot update the PricebookId field or the CurrencyIsoCode field on the opportunity if line items exist. The API rejects any attempt to update these fields on an opportunity with line items.
- When you Create() or Update() an OpportunityLineItem, the API verifies that the line item corresponds to a PricebookEntry in the Pricebook2 that is associated with the opportunity. If the opportunity does not have an associated Pricebook2, the API automatically sets the pricebook on the opportunity if the line item corresponds to a PricebookEntry in an active Pricebook2, and if the PricebookEntry has a CurrencyIsoCode field that matches the CurrencyIsoCode field of the opportunity. If the Pricebook2 is not active or the CurrencyIsoCode fields do not match, an error is returned.
- The Opportunity HasOpportunityLineItem field is set to true when an OpportunityLineItem is inserted for that Opportunity.

OpportunityLineItemSchedule

Represents information about the quantity, revenue distribution, and delivery dates for a particular OpportunityLineItem.

Corresponds to an SObject4 in which the ObjectType="OpportunityLineItemSchedule".

Supported Calls

```
Create(),Update(),DescribeSObjects(),Query(),Retrieve(),CreateObject(),GetDeleted(),
GetUpdated()
```

Field	Field Type	Field Properties	Description
CurrencyIsoCode	picklist	Create Defaulted on create	Available only for organizations with the multi-currency feature enabled. Contains the ISO code for any currency allowed by the organization. This field is available in version
		Filter	10.0 and later.
		Nillable	
		Restricted picklist	
		Update	
Description	string	Create	Text description of the opportunity line item schedule. Limit:
		Filter	255 characters. Label is Comments .
		Nillable	

Field	Field Type	Field Properties	Description
		Update	
IsDeleted	boolean	Defaulted on create Filter	Indicates whether the record has been moved to the Recycle Bin (true) or not (false). Label is Deleted .
OpportunityLineItemId	reference	Create Filter	Required. ID of the associated OpportunityLineItem.
Quantity	double	Create Filter Nillable Update	Required. The total number of units to be scheduled in a quantity schedule. See Allowed Quantity and Revenue Field Values for more information.
Revenue	currency	Create Filter Nillable Update	The revenue that should be recognized, or the quantity that should be shipped, or both - depending upon the value of T_{ype} . See Allowed Quantity and Revenue Field Values for more information.
ScheduleDate	date	Create Filter Update	Required. The date the associated OpportunityLineItem is to be scheduled for an event: delivery, shipping, or any other date you wish to track.
Туре	picklist	Create Filter Restricted picklist	Required. The type of the schedule. Required when inserting an OpportunityLineItemSchedule. Valid values include Quantity, Revenue, or Both. See Allowed Type Field Values for more information.

Allowed Type Field Values

The allowed T_{ype} values for an OpportunityLineItemSchedule depend on the product-level schedule preferences and whether the line item has any existing schedules. The following criteria must be met:

- The Product2 on which the OpportunityLineItem is based must have the appropriate CanUseRevenueSchedule or CanUseQuantitySchedule fields (or both) set to true.
- When you Create() a schedule for a line item that does not have any existing schedules, you can specify any valid value.
- If you Create() a schedule for a line item that already has existing schedules, the new schedule must be consistent with the existing schedules. The following matrix outlines the allowable values:

Value of HasRevenueSchedule on line item	Value of HasQuantitySchedule on line item	Allowable Type Values
false	false	Revenue, Quantity, both
false	true	Quantity

Value of HasRevenueSchedule on line item	Value of HasQuantitySchedule on line item	Allowable Type Values
true	false	Revenue
true	true	both

Allowed Quantity and Revenue Field Values

The allowable Quantity and Revenue field values depend on the value of the Type field:

Type Value	Allowable Quantity Value	Allowable Revenue Value
Revenue	Null	Non-null
Quantity	Non-null	Null
both	Non-null	Non-null

The Quantity and Revenue fields have the following restrictions in the Update () call:

- For a schedule of Type "Quantity," you cannot update a null Revenue value to non-null. Likewise for a schedule of Type "Revenue," you cannot update a null Quantity value to non-null.
- You cannot null out the Quantity field for a schedule of Type "Quantity." Likewise you cannot null out the Revenue field for a schedule of Type "Revenue."
- You cannot null out either the Revenue or Quantity fields for a schedule of type "Both."

Usage

Two types of OpportunityLineItemSchedules are supported:

- Quantity schedules
- Revenue schedules

The user must have edit access rights on the Opportunity in order to Create() or Update() line item schedules on that opportunity.

Products and Schedules Must Be Enabled

The OpportunityLineItemSchedule object is defined only for those organizations that have the products and schedules features enabled. If the organization does not have the products and schedules features, the OpportunityLineItemSchedule object does not appear in the DescribeGlobal() call, and you cannot use DescribeSObjects() or Query() with the OpportunityLineItemSchedule object.

Effects on Opportunities and Opportunity Line Items

OpportunityLineItemSchedules affect opportunities and opportunity line items in the following ways:

- Inserting an OpportunityLineItemSchedule of Type "Revenue" or "Quantity" increments the TotalPrice field on the OpportunityLineItem by the OpportunityLineItemSchedule Revenue amount. Inserting an OpportunityLineItemSchedule of Type "Quantity" or "Both" increments the Quantity field on the OpportunityLineItem by the OpportunityLineItemSchedule Quantity amount.
- The Create() call also affects the original opportunity:

1. The Opportunity Amount is incremented the by OpportunityLineItemSchedule revenue amount

- 2. The Opportunity ExpectedRevenue is incremented by the line item schedule amount multiplied by the Opportunity Probability
- Deleting an OpportunityLineItemSchedule has a similar effect on the related OpportunityLineItem and Opportunity. Deleting an OpportunityLineItemSchedule decrements the OpportunityLineItem TotalPrice by the deleted OpportunityLineItemSchedule Quantity or Revenue amount. The Opportunity Amount is also decremented by the OpportunityLineItemSchedule Quantity or Revenue amount, and the Opportunity ExpectedRevenue is reduced by OpportunityLineItemSchedule Quantity or Revenue amount multiplied by the Opportunity Probability.

Deleting an Opportunity Line Item Schedule

Deleting the last remaining schedule will set the corresponding HasQuantitySchedule or HasRevenueSchedule flags (or both) to false on the parent line item.

OpportunityOverride

A forecast override of an Opportunity.

Supported Calls

Query(),Retrieve(),GetDeleted(),GetUpdated(),DescribeSObjects()

Special Access Rules

Requires the "View All Data" permission.

Field	Field Type	Field Properties	Description
AmountInherited	boolean	Defaulted on create Filter	Indicates whether the overridden amount rolls up through the forecast hierarchy (true), or was overridden by the owner of the OpportunityOverride (false).
ForecastCategoryInherited	boolean	Defaulted on create Filter	Indicates whether the overridden forecast category rolls up through the forecast hierarchy (true) or was overridden by the owner of the OpportunityOverride (false).
IsDeleted	boolean	Defaulted on create Filter	Indicates whether the object has been moved to the Recycle Bin (true) or not (false), usually because the parent Opportunity is moved to the Recycle Bin.
OpportunityId	reference	Filter	ID of the associated Opportunity. For information on IDs, see ID Field Type.
OutOfDate	boolean	Defaulted on create Filter	true if a subordinate user's OpportunityOverride, or the opportunity itself, has been updated since this override was last updated, such that an overridden value on this override may be obsolete. For example, a subordinate user has more

Field	Field Type	Field Properties	Description
			recently overridden the same period, category, or amount field.
OverrideAmount	currency	Filter	The total monetary amount of the opportunity, possibly overridden.
OverrideComment	string	Filter	The comment entered on the opportunity forecast edit page.
		Nillable	Limit: 255 characters.
OverrideForecastCategory	picklist	Filter	The forecast category of the opportunity, possibly overridden.
		Restricted picklist	
OverridePeriodId	reference	Filter	ID of the associated fiscal Period, possibly overridden. If you are using custom fiscal years and the period falls after your custom fiscal year has ended, then the override is not returned. For information on custom fiscal years, see the Salesforce online help.
OverrideQuantity	double	Filter	The quantity of the opportunity, possibly overridden.
OwnerId	reference	Filter	ID of the OpportunityOverride owner.
PeriodInherited	boolean	Defaulted on create	Indicates whether the overridden period rolls up through the forecast hierarchy (true) or was overridden by the
		Filter	OpportunityOverride owner (false).
QuantityInherited	boolean	Defaulted on create Filter	Indicates whether the overridden quantity rolls up through the forecast hierarchy (true) or was overridden by the OpportunityOverride owner (false).

This is read-only object specific to customizable forecasting. It has a parent-child relationship with LineitemOverride.

OpportunityOwnerSharingRule

Represents the rules for sharing an opportunity with users other than the owner.

Note: Contact salesforce.com customer support to enable access to this object for your organization.

Supported Calls

```
Create(),Update(),DescribeSObjects(),Query(),Retrieve(),CreateObject(),GetDeleted(),
GetUpdated(),DescribeSObjects()
```

Use these objects to manage the sharing rules for a particular opportunity. General sharing uses this object.

OpportunityPartner

This read-only object represents a partner relationship between an Account and an Opportunity. This object is automatically created when a Partner object is created for a partner relationship between an account and an opportunity.

Supported Calls

Query(),Retrieve(),DescribeSObjects()

Fields

Field	Field Type	Field Properties	Description
AccountToId	reference	Filter	ID of the partner Account in the partner relationship. For information on IDs, see ID Field Type.
IsDeleted	boolean	Defaulted on create Filter	Indicates whether the record has been moved to the Recycle Bin (true) or not (false). Label is Deleted .
IsPrimary	boolean	Defaulted on create Filter	Indicates whether the account is the opportunity's primary partner (true) or not (false). Label is Primary .
OpportunityId	reference	Filter	ID of the Opportunity that is in the partner relationship.
Role	picklist	Filter Nillable	The UserRole that the Account has on the Opportunity. For example, "Reseller" or "Manufacturer."

Creating an Account-Opportunity Partner Relationship

When you create a partner relationship between an account and an opportunity (when you create a Partner object and specify the OpportunityId field), the API automatically creates an OpportunityPartner with the corresponding values:

- The value of the Partner field AccountToId maps to the value of the OpportunityPartner field AccountToId.
- The values of the OpportunityId, Role, and IsPrimary fields in both objects are the same.
- If you set the IsPrimary value to 1 (true) upon insert of a new OpportunityPartner, any other existing primary partners for that opportunity will automatically have the IsPrimary value set to 0 (false).

This mapping allows the API to manage the objects and their relationship efficiently.

OpportunityShare

Represents a sharing entry on an Opportunity.

Corresponds to an SObject4 in which the ObjectType="OpportunityShare".

Supported Calls

Create(),Update(),DescribeSObjects(),Query(),Retrieve(),CreateObject()

Field	Field Type	Field Properties	Description
IsDeleted	boolean	Defaulted on create	Indicates whether the record has been moved to the Recycle Bin (true) or not (false). Label is Deleted .
		Filter	
OpportunityAccessLevel	picklist	Create Filter Restricted picklist Update	<pre>Level of access that the user or group has to the opportunity. The possible values are: Read Edit All (This value is not valid for delete(), Create(), or Update() calls.)</pre>
			This field must be set to an access level that is higher than the organization's default access level for opportunities.
OpportunityId	reference	Create Filter	ID of the Opportunity associated with this sharing entry. This field cannot be updated. For information on IDs, see ID Field Type.
RowCause	picklist	Filter Restricted picklist	 Reason that this sharing entry exists. Read-only. Valid values include: Owner—User is the owner of the opportunity or is in a UserRole above the opportunity owner in the role hierarchy. Manual—User or group has access because a user with "All" access manually shared the opportunity with them. Rule—User or group has access via an opportunity sharing rule. ImplicitChild—User or group has access to the opportunity. Team—User has access to the opportunity because she or he is on the sales team for the opportunity. The OpportunityTeamMember object for this opportunity sets the access level. See OpportunityTeamMember for more information.
UserOrGroupId	reference	Create Filter	ID of the User or Group that has been given access to the opportunity. This field cannot be updated.

This object allows you to determine which users and groups can view or edit opportunities owned by other users. For more information, see Sharing.

If you attempt to create a record that matches an existing record, the Create() call updates any modified fields and returns the existing record.

OpportunityStage

Represents the stage of an Opportunity in the sales pipeline, such as New Lead, Negotiating, Pending, Closed, and so on.

Corresponds to an SObject4 in which the ObjectType="OpportunityStage".

Supported Calls

Query(),Retrieve(),CreateObject(),DescribeSObjects()

Field	Field Type	Field Properties	Description
DefaultProbability percent	percent	Filter	The default percentage estimate of the confidence in closing
		Nillable	a specific opportunity for this opportunity stage value. Label is Probability (%) .
Description	string	Filter	Description of this opportunity stage value. Limit: 255
	Nillable	characters.	
ForecastCategory	picklist	Filter	The default forecast category for this opportunity stage value.
	Restricted picklist	The forecast category automatically determines how opportunities are tracked and totaled in a forecast.	
ForecastCategoryName	picklist	Filter	Available in API version 12.0 and later. The default forecast category value for this opportunity stage value.
IsActive I	boolean	Defaulted on create	Indicates whether this opportunity stage value is active (true) or not (false). Inactive opportunity stage values
		Filter	are not available in the picklist and are retained for historical purposes only.
IsClosed	boolean	Defaulted on create	Indicates whether this opportunity stage value represents a closed opportunity (true) or not (false). Multiple
F	Filter	opportunity stage values can represent a closed opportunity. Label is Closed .	
IsWon	boolean	Defaulted	Indicates whether this opportunity stage value represents a M_{1}
		Filter	opportunity (crue) of not (rarse). Multiple opportunity stage values can represent a won opportunity. Label is Won .

Field	Field Type	Field Properties	Description
MasterLabel	string	Filter Nillable	Master label for this opportunity stage value. This display value is the internal label that does not get translated. Limit: 255 characters.
SortOrder	int	Filter Nillable	Number used to sort this value in the opportunity stage picklist. These numbers are not guaranteed to be sequential, as some previous opportunity stage values might have been deleted.

This object represents a value in the opportunity stage picklist, which provides additional information about the stage of a Opportunity, such as its probability or forecast category. Your client application can invoke the Query () call on the OpportunityStage object to retrieve the set of values in the opportunity stage picklist, and then use that information while processing Opportunity objects to determine more information about a given opportunity. For example, the application could test whether a given opportunity is won or not based on its StageName value and the value of the IsWon property in the associated OpportunityStage object.

This object is read-only via the API. With sufficient permissions, your client application can invoke the Query () and DescribeSObjects() calls on these objects.

OpportunityTeamMember

Represents a User on the sales team of an Opportunity.

Corresponds to an SObject4 in which the ObjectType="OpportunityTeamMember".

See also UserTeamMember, which represents a User who is on the default sales team of another user.

Supported Calls

```
Create(),Update(),DescribeSObjects(),Query(),Retrieve(),CreateObject(),GetDeleted(),
GetUpdated()
```

Field	Field Type	Field Properties	Description
IsDeleted	boolean	Defaulted on create Filter	Indicates whether the record has been moved to the Recycle Bin (true) or not (false). Label is Deleted .
OpportunityAccessLevel	picklist	Filter Nillable Restricted picklist	Opportunity access level for this team member. The possible values are: • None • Read • Edit • All

Field	Field Type	Field Properties	Description
OpportunityId	reference	Create Filter	Required. ID of the Opportunity associated with this sales team. This field cannot be updated. For information on IDs, see ID Field Type.
TeamMemberRole	picklist	Create Filter Nillable Update	Role that the team member has on the opportunity. The valid values are set by the organization's administrator in the Sales Team Roles picklist. Label is Team Role .
UserId	reference	Create Filter	Required. ID of the User who is a member of the Opportunity's sales team. This field cannot be updated.

If you attempt to create this object and it matches an existing record, the Create() call updates any modified fields and returns the existing record.

In the Salesforce user interface, users can set up a sales team for the opportunities they own. The sales team includes other users that are working on the opportunity with them. This object is available only in organizations that have enabled the team selling functionality.



Note: The behavior for changing ownership of opportunities is different using the Salesforce user interface when the previous owner is on a sales team. For example, when you change the owner of an opportunity using the API, the previous owner's access becomes Read Only or the access specified in your organization-wide default for opportunities, whichever is greater. However, performing this same action in Salesforce allows you to select the access level for the previous owner when he or she is on a sales team.

OpportunityTag

Associates a word or short phrase with an Opportunity.

Supported Calls

```
Create(),Query(),Retrieve(),DescribeSObjects()
```

Field	Field Type	Field Properties	Description
ItemId	reference	Create Filter	ID of the tagged item.
Name	string	Create Filter	Name of the tag. If this value does not already exist, a new TagDefinition is created and becomes the parent of this Tag object. Otherwise, a TagDefinition with the same name

Field	Field Type	Field Properties	Description
			becomes the parent of this Tag object. Parent relationships are created automatically.
TagDefinitionId	reference	Filter	ID of the parent TagDefinition object that owns the tag.
Туре	picklist	Create Filter Restricted picklist	 Defines the visibility of a tag. Possible value are: Public: The tag can be viewed and manipulated by all users in an organization Personal: The tag can be viewed or manipulated only by a user with a matching OwnerId

OpportunityTag stores the relationship between its parent TagDefinition and the Opportunity being tagged. Tag objects act as metadata, allowing users to describe and organize their data.

When a tag is deleted, its parent TagDefinition will also be deleted if the name is not being used; otherwise, the parent remains. Deleting a TagDefinition sends it to the recycle bin, along with any associated tag entries.

For more information on tags, see "About Tagging" in the Salesforce online help.

Order

Represents a shipment or provisioning of products, an order, that is associated with an account, contract, and opportunity. This object provides a way to create orders and track them, and is also an interface for integrating with another ERP or order management system.

This object is available to organizations with the orders feature enabled.

Supported Calls

```
Create(),Update(),Query(),Retrieve(),CreateObject(),GetDeleted(),GetUpdated(),
DescribeSObjects()
```

Field	Field Type	Field Properties	Description
AccountId	reference	Filter	ID of the account associated with this order. This ID is the same as the account ID associated with the specified contract ID.
ActivatedById	reference	Filter Nillable	ID of the user who activated this order.
ActivatedDate	dateTime	Filter	Date and time when this order was activated.

Field	Field Type	Field Properties	Description
• BillingCity	string	Create	Details of the billing address.
• BillingCountry		Filter	
 BillingPostalCode BillingState 		Nillable	
		Update	
BillingStreet	textarea	Create	Details of the billing street for the billing address.
		Filter	
		Nillable	
		Update	
CompanyAuthorizedById	reference	Create	ID of the user who signed the order.
		Filter	
		Nillable	
		Update	
ContractID	reference	Create	Required. The specified contract must be associated with a
		Filter	Pricebook21d. The price book entries that can be added as s are determined by the Status and the CurrencyIsoCode value set on the Contract.
CustomerAuthorizedById	reference	Create	ID of the contact who signed the order.
		Filter	
		Nillable	
		Update	
Description	textarea	Create	Description of the order.
		Nillable	
		Update	
EffectiveDate	date	Create	Required. Represents the Order Start Date field in the online
		Filter	if one is specified. Also, it must be on or before the Contract Start Date ,
		Update	End Date, if one is specified.
IsDeleted	boolean	Defaulted on create	Indicates whether the record has been moved to the Recycle Bin (true) or not (false). Label is Deleted .
OrderNumber	string	Autonumber	Number of the order.
		Defaulted on create	
		Filter	

Field	Field Type	Field Properties	Description
 ShippingCity ShippingCountry ShippingPostalCode ShippingState 	string	Create Filter Nillable Update	Details of the shipping address.
ShippingStreet	textarea	Create Filter Nillable Update	Details of the shipping street for the shipping address.
Status	picklist	Create Filter Update	Required. The picklist of values that indicate order status. Each value is within one of two status categories defined in StatusCode, Draft or Activated. For example, the status picklist may contain: Ready to Ship, Shipped, Received as values within the Activated StatusCode.
StatusCode	picklist	Filter Restricted picklist	The status category for the order. An order can be either Draft or Activated.
TotalAmount	currency	Filter	Read only. Its value is determined by the sum of the order's OrderItem records, if any exist. It equals the sum of the UnitPrice fields on the OrderItem records.
Туре		Create Filter	The type of order, for example, new order or upgrade order.
		Nillable Update	

Use this object to create, activate, edit, or de-activate orders.

This object supports cascading deletes for notes, attachments, and activities. It supports custom fields.

OrderHistory

Represents the history of an order.

Supported Calls

Query(),Retrieve(),CreateObject(),GetDeleted(),GetUpdated(),andDescribeSObjects()

Special Access Rules

This object is available to organizations with the orders feature enabled.

Fields

Field	Field Type	Field Properties	Description
Field	picklist	Filter Restricted picklist	 Name of the order field that was modified, or a special value to indicate some other modification to the order. The possible values include: BillingAddress, Billing City, BillingCountry, BillingPostalCode, BillingState, BillingStreet from Order. BillToContact CompanyAuthorizedBy from Order, CompanyAuthorizedDate Created. CustomerAuthorizedBy, CustomerAuthorizedDate Description EffectiveDate Name Order activated, Order approved, Order canceled, Order deactivated, Order submitted, OrderReferenceNumber Owner (Accepted), Owner (Assignment) PoDate, PoNumber Record locked, Record unlocked ShippingAddress, ShippingCity, ShippingStreet ShipToContact Status Type Label is Custom Field Definition ID.
IsDeleted	boolean	Defaulted on create Filter	Indicates whether the object has been moved to the Recycle Bin (true) or not (false). Label is Deleted .
OrderId	reference	Filter	ID of the Order associated with this record. For information on IDs, see ID Field Type. Label is Order ID.

Usage

Query this read-only object to describe the history of an Order.

This object respects field level security on the parent object.

OrderItem

Represents a single item in a shipment or provisioning of products.

This object is available to organizations with the orders feature enabled.

Supported Calls

Create(),Update(),DescribeSObjects(),Query(),Retrieve(),CreateObject(),GetDeleted(), GetUpdated()

Fields

Field	Field Type	Field Properties	Description
IsDeleted	boolean	Defaulted on create	Indicates whether the record has been moved to the Recycle Bin (true) or not (false). Label is Deleted .
		Filter	
ListPrice	currency	Filter	List price for the order item.
		Nillable	
OrderId	reference	Create	ID of the Order that is associated with this record.
		Filter	
PricebookEntryId	reference	Create	ID of the PricebookEntry that is associated with this record.
		Filter	
Quantity	double	Create	The quantity of this order item in the order.
		Filter	
		Update	
UnitPrice	currency	Create	The price charged for one of this item. Unit prices for all
		Filter	OrderItem in an Order are calculated and populate the Order field TotalAmount.
		Nillable	
		Update	

Usage

Use this object to describe a single item and associate it with an Order.

OrderTag

Associates a word or short phrase with an Order.

Supported Calls

Create(),Query(),Retrieve(),DescribeSObjects()

Fields

Field	Field Type	Field Properties	Description
ItemId	reference	Create Filter	ID of the tagged item.
Name	string	Create Filter	Name of the tag. If this value does not already exist, a new TagDefinition is created and becomes the parent of this Tag object. Otherwise, a TagDefinition with the same name becomes the parent of this Tag object. Parent relationships are created automatically.
TagDefinitionId	reference	Filter	ID of the parent TagDefinition object that owns the tag.
Туре	picklist	Create Filter Restricted picklist	 Defines the visibility of a tag. Possible value are: Public: The tag can be viewed and manipulated by all users in an organization Personal: The tag can be viewed or manipulated only by a user with a matching OwnerId

Usage

OrderTag stores the relationship between its parent TagDefinition and the Order being tagged. Tag objects act as metadata, allowing users to describe and organize their data.

When a tag is deleted, its parent TagDefinition will also be deleted if the name is not being used; otherwise, the parent remains. Deleting a TagDefinition sends it to the recycle bin, along with any associated tag entries.

For more information on tags, see "About Tagging" in the Salesforce online help.

Organization

This object represents key configuration information for an organization. You must have the "View All Data" permission to access this object.

Supported Calls

GetUpdated()

Executing a SOQL SELECT query returns the value of fields in this object, but no value is visible for some of the fields.

Special Access Rules

Customer Portal users cannot access this object.

Field	Field Type	Field Properties	Description
AllowsSelfServiceLogin	boolean	Defaulted on create Filter	Indicates whether the organization allows Self-Service login (true) or not (false).
		Opdate	
City	string	Filter	Name of the city for the organization's address.
		Nillable	
		Update	
ComplianceBccEmail	email	Filter	Email address for compliance blind carbon copies.
		Nillable	Limit: 80 characters.
Country	string	Filter	Name of the country for the organization's address.
		Nillable	
DailyWebToCaseCount	int	Filter	The number of Web form submissions that have been
		Nillable	converted to cases for the day.
DailyWebToCaseLimit	int	Filter	The maximum number of Web form submissions that
		Nillable	can be converted to cases per day.
DailyWebToLeadCount	int	Filter	The number of Web form submission that have been
		Nillable	converted to leads for the day.
DailyWebToLeadLimit	int	Filter	The maximum number of Web form submissions that
		Nillable	can be converted to leads per day.
DefaultAccountAccess	picklist	Filter	In API version 10.0 and later, represents the default
		Nillable	access level for accounts, contracts, and assets. The
		Restricted	None
		picklist	• Read
			• Edit
			In versions before 10.0,
			this value.
DefaultAccountAndContactAccess	picklist	Filter	Default access level for accounts, contacts, contracts,
		Nillable	and assets. This field is supported for backward
		Restricted	10.0 or later. In version 10.0 and later, use either
		picklist	DefaultAccountAccess or DefaultContactAccess.

Field	Field Type	Field Properties	Description
DefaultCalendarAccess	picklist	Filter Nillable Restricted picklist	 Default access level for calendars. The possible values are listed, followed by the Salesforce user interface labels in parentheses: HideDetails (Hide Details) HideDetailsInsert (Hide Details and Add Events) ShowDetails (Show Details) ShowDetailsInsert (Show Details and Add Events) AllowEdits (Full Access)
DefaultCampaignAccess	picklist	Filter Nillable Restricted picklist	<pre>Default access level for campaigns. The possible values are: NoneRead Edit All</pre>
DefaultCaseAccess	picklist	Filter Nillable Restricted picklist	 Default access level for cases. The possible values are: NoneRead Edit ReadEditTransfer
DefaultContactAccess	picklist	Filter Nillable Restricted picklist	Default access level for contacts. The possible values are: • None • Read • Edit • ControlledByParent In versions before 10.0, DefaultAccountAndContactAccess represented this value. Note: When DefaultContactAccess is set to "Controlled by Parent," you cannot update an object's ContactAccessLevel field.
DefaultLeadAccess	picklist	Filter Nillable Restricted picklist	<pre>Default access level for leads. The possible values are: NoneRead Edit ReadEditTransfer</pre>
DefaultLocaleSidKey	picklist	Filter Restricted picklist	Default locale SID key.

Field	Field Type	Field Properties	Description
DefaultOpportunityAccess	picklist	Filter Restricted picklist	Default access level for opportunities. The possible values are: • None • Read • Edit
DefaultPricebookAccess	picklist	Filter Restricted picklist	<pre>Default access level for price books. The possible values are listed, followed by the Salesforce user interface labels in parentheses: None (No access) Read (Read only) ReadSelect (Use)</pre>
DefaultTerritoryAccountAccess	picklist	Filter Nillable Restricted picklist	 Default access level for accounts in territories. The possible values are: Read Edit All
DefaultTerritoryCaseAccess	picklist	Filter Nillable Restricted picklist	 Default access level for cases associated with accounts in territories. The possible values are: None Read Edit
DefaultTerritoryContactAccess	picklist	Filter Nillable Restricted picklist	 Default access level for contacts associated with accounts in territories. The possible values are: NoneRead Edit Note: When DefaultContactAccess is set to "Controlled by Parent" you cannot update this field.
DefaultTerritoryOppAccess	picklist	Filter Nillable Restricted picklist	Default access level for opportunities in territories. The possible values are:NoneReadEdit
Division	string	Filter Nillable Update	The name of the division for this organization. This field is not related to the Division object.
Fax	phone	Filter Nillable	Fax number. Limit: of 40 characters.

Field	Field Type	Field Properties	Description
		Update	
FiscalYearStartMonth	int	Filter	Number that corresponds to the month that this
		Nillable	organization's fiscal year starts.
HomepageHtml	textarea	Nillable	The Home tab custom links and company message for this organization.
		Update	5
LanguageLocaleKey	picklist	Filter	Language locale key.
		Restricted picklist	
		Update	
LastWebToCaseDate	dateTime	Filter	The last date that a Web form submission was converted
		Nillable	to a case.
LastWebToLeadDate	dateTime	Filter	The last date that a Web form submission was converted
		Nillable	to a lead.
MaxActionsPerRule	int	Filter	Maximum number of actions per workflow, assignment,
		Nillable	This field is upgraitable in version 15.0 and later
			This field is unavailable in version 15.0 and later.
MaxRulesPerEntity	int	Filter Nillable	Maximum number of rules per object, inclusive of workflow, assignment, escalation, and auto-response rules.
			This field is unavailable in version 15.0 and later.
Name	string	Filter	The name of the organization.
		Update	
OrganizationType	picklist	Filter	Edition of the organization, for example Enterprise
		Nillable	Edition or Unlimited Edition.
Phone	phone	Filter	Phone number for the organization.
		Nillable	
		Update	
PostalCode	string	Filter	Postal code for the address of the organization. Limit:
		Nillable	20 characters.
		Update	
PreferencesEventScheduler	boolean	Update	Indicates whether opportunities require products (true) or not (false).

Field	Field Type	Field Properties	Description
PreferencesRequireOpportunityPro ducts	boolean	Update	Indicates whether opportunities require products (true) or not (false).
PrimaryContact	string	Filter	Name of the primary contact for the organization. Limit:
		Nillable	of 80 characters.
		Update	
ReceivesAdminInfoEmails	boolean	Defaulted on create	Indicates whether the organization receives Salesforc administrator emails (true) or not (false).
		Filter	
		Update	
ReceivesInfoEmails	boolean	Defaulted on create	Indicates whether the organization receives informational email from Salesforce (true) or not
		Filter	(false).
		Update	
SelfServiceCasePlural	string	Filter	The plural version of the term used to represent the
		Nillable	Case object in the Self-Service portal.
		Update	
SelfServiceCaseSingle	string	Filter	The singular version of the term used to represent the
		Nillable	Case object in the Self-Service portal.
		Update	
SelfServiceCaseSubmitRecordTypeId	reference	Filter	The ID of the record type associated with a case
		Nillable	submitted via the Self-Service portal.
		Update	
SelfServicDefaultCaseOrigin	string	Filter	The default origin of a case submitted via the
		Nillable	Self-Service portal.
		Update	
SelfServiceEmailSenderAddress	email	Filter	The Self-Service email address from which new
		Nillable	Self-Service user and password email messages are sent,
		Update	
SelfServiceEmailSenderName	string	Filter	The name associated with the email address in the SelfServiceEmailSenderAddress field, such as "Acme Customer Support."
		Nillable	
		Update	
SelfServiceEmailUserOnCase CreationTemplateId	reference	Filter Nillable Update	The ID of the email template used when email is sent to a Self-Service user when he or she creates a case.

Field	Field Type	Field Properties	Description
SelfServiceEnabledForResponseRules	boolean	Filter	Indicates whether the Self-Service portal is enabled for auto-response rules (true) or not (false).
		Nillable	1
		Update	
SelfServiceFeatureConfig	int	Filter Nillable Update	An integer representing the active Self-Service feature configuration for this organization.
SelfServiceLogoutUrl	url	Filter	The Web page that displays when a Self-Service use
		Nillable	logs out of the Self-Service portal.
		Update	
SelfServiceMaxNumSuggestions	int	Filter	The maximum number of suggested solutions allowed
		Nillable	for a Self-Service case.
		Update	
SelfServiceNewCommentCheckedByDe fault	boolean	Defaulted on create	Indicates whether the checkbox to send customer notifications whenever there is a new case comment
		Filter	automatically selected.
		Update	
SelfServiceNewCommentTemplateId	reference	Filter	The ID of the email template used to send a notification
		Nillable	to Self-Service users when a public comment is added
		Update	to one of their cases.
SelfServiceNewPassTemplateId	reference	Filter	The ID of the email template used when new passwords
		Nillable	are generated for Self-Service users.
		Update	
SelfServiceNewUserTemplateId	reference	Filter	The ID of the email template used when new
		Nillable	Self-Service users are enabled.
		Update	
SelfServicePageHeight	int	Filter	The maximum height in pixels of Self-Service pages.
		Nillable	
		Update	
SelfServicePageWidth	int	Filter	The maximum width in pixels of Self-Service pages.
		Nillable	
		Update	
SelfServiceSelfClosedCaseStatus	picklist	Filter	The default status for cases closed by Self-Service users.
		Nillable	

Field	Field Type	Field Properties	Description
		Update	
SelfServiceSolutionCategoryAvailable	boolean	Defaulted on create	Indicates whether solution categories are available in the Self-Service portal (true) or not (false).
		Filter	
		Update	
SelfServiceSolutionCategorySta	reference	Filter	The ID of the top-level category in the Self-Service
rtNodeId		Nillable	portal.
		Update	
SelfServiceSolutionPlural	string	Filter	The plural version of the term used to represent the
		Nillable	Solution object in the Self-Service portal.
		Update	
SelfServiceSolutionSingle	string	Filter	The singular version of the term used to represent the
		Nillable	Solution object in the Self-Service portal.
		Update	
SelfServiceStyleSheetUrl	url	Filter	The public URL of your organization's Self-Service
		Nillable	portal stylesheet.
		Update	
SelfServiceWelcomePageConfig	int	Filter	Integer that represents the welcome page configurat
		Nillable	for the Self-Service portal.
		Update	
SelfServiceWelcomeText	string	Filter	The custom welcome message displayed at the top of
		Nillable	the Self-Service home page when Self-Service users log in. Limit: 32,000 characters.
		Update	
State	string	Filter	State of the address of the organization. Limit: 20
		Nillable	characters.
		Update	
Street	textarea	Filter	Street address for the organization. Limit: 255
		Nillable	characters.
		Update	
TrialExpirationDate	dateTime	Filter	The date that this organization's trial license expires.
		Nillable	

Field	Field Type	Field Properties	Description
UiSkin	picklist	Filter	The user interface theme selected for the organization.
		Nillable	
		Restricted picklist	
UsesStartDateAsFiscalYearName	boolean	Defaulted	Indicates whether the calendar year when the fiscal year
		Gil cicate	year (true) or not (false). For example, if the fiscal
		FILCI	year begins in February 2006, a true value means the fiscal year is FY2006, and a false value means the fiscal year is FY2007.
UsesWebToCase	boolean	Filter	Indicates whether this organization can use
		Nillable	Web-to-Case (true) or not (false).
		Update	
UsesWebToLead	boolean	Filter	Indicates whether this organization can use
		Nillable	Web-to-Lead (true) or not (false).
		Update	
WebToCaseAssignedEmailTemplateId	reference	Filter	The ID of the email template used when a new case is
		Nillable	assigned to a user via Web-to-Case.
		Update	
WebToCaseCreatedEmailTemplateId	reference	Filter	The ID of the email template used when a new case is
		Nillable	created via Web-to-Case.
		Update	
WebToCaseDefaultCreatorId	reference	Filter	The ID of the user specified as the default creator of
		Nillable	cases created via Web-to-Case.
		Update	
WebToCaseDefaultOrigin	string	Filter	The default value for the Case Origin field on cases
		Nillable	submitted via Web-to-Case. Limit: 40 characters.
		Update	

Query this object to obtain information about an organization's settings. Only one organization object exists per organization.

Partner

Represents a partner relationship between two Accounts or between an Opportunity and an Account.

Corresponds to an SObject4 in which the ObjectType="Partner".



Note: This object is completely independent of and distinct from PRM Portal functionality.

Supported Calls

```
Create(),DescribeSObjects(),Query(),Retrieve(),CreateObject(),GetDeleted(),GetUpdated()
```

Special Access Rules

- You must have the "View All Data" permission to access this object via the API. All of the Partner fields are accessible in the DescribeSObjects() and Query() call. You cannot Update() partners via the API.
- Customer Portal users cannot access this object.

Field	Field Type	Field Properties	Description
AccountFromId	reference	Create Filter Nillable	Required if OpportunityId is null. ID of the main Account in a partner relationship between two accounts. Specifying this field when calling Create() creates two AccountPartner objects, one for each direction of the relationship. If you specify the OpportunityId field, you cannot specify this field as well. For information on IDs, see ID Field Type.
AccountToId	reference	Create Filter	Required. ID of the Partner Account related to either an opportunity or an account. You must specify this field when creating an opportunity Partner or an Account Partner.
IsDeleted	boolean	Defaulted on create Filter	Indicates whether the record has been moved to the Recycle Bin (true) or not (false). Label is Deleted .
IsPrimary	boolean	Create	Valid for Opportunity Partners only.
		Defaulted on create Filter	Indicates that the account is the primary partner for the opportunity. Only one account can be marked as primary for an opportunity. If you set this field to 1 (true) upon insert of a new opportunity partner, any other primary partners for that opportunity will automatically have this field set to 0 (false). Label is Primary .
Field	Field Type	Field Properties	Description
---------------	------------	------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
OpportunityId	reference	Create Filter Nillable	Required if AccountFromId is null. ID of the Opportunity in a partner relationship between an Account and an Opportunity. Specifying this field when calling Create() creates an OpportunityPartner. If you specify the AccountFromId field, you cannot specify this field as well.
Role	picklist	Create Filter Nillable	UserRole that the account has towards the related opportunity or account (for example, consultant or distributor).

Roles

In the Salesforce user interface, system administrators can set up the valid role values and their corresponding reverse role values in the PartnerRole object. Each account in the relationship is assigned a Role (such as "Consultant" or "Distributor") designating that account's role towards the related account or opportunity.

Creating an Account-Opportunity Partner Relationship

When you create a partner relationship between an account and an opportunity (when you create a Partner object and specify the OpportunityId field), the API automatically creates an OpportunityPartner with the corresponding values:

- The value of the Partner field AccountToId maps to the value of the OpportunityPartner field AccountToId.
- The values of the OpportunityId, Role, and IsPrimary fields in both objects are the same.
- If you set the IsPrimary value to 1 (true) upon insert of a new OpportunityPartner, any other existing primary partners for that opportunity will automatically have the IsPrimary value set to 0 (false).

This mapping allows the API to manage the objects and their relationship efficiently.

Creating an Account-Account Partner Relationship

When you create a partner relationship between two accounts (when you create a Partner object and specify the AccountFromId), the API automatically creates two AccountPartner objects, one for the forward relationship and one for the reverse. For example, if you create a Partner object with "Acme, Inc." as the AccountFromId and "Acme Consulting" as the AccountToId, the API automatically creates two AccountPartner objects:

- The forward relationship AccountPartner with "Acme, Inc." as the AccountFromId and "Acme Consulting" as the AccountToId.
- The reverse relationship AccountPartner with "Acme Consulting" as the AccountFromId and "Acme, Inc." as the AccountToId.
- The value of the Role field in the reverse relationship AccountPartner is set to the PartnerRole object ReverseRole value associated with the value of the Role field in the forward relationship AccountPartner.

This mapping allows the API to manage the objects and their relationship efficiently.

PartnerNetworkConnection

Represents a Salesforce to Salesforce connection between Salesforce organizations.

Supported Calls

Query(),Retrieve(),DescribeSObjects()

Fields

Field	Field Type	Field Properties	Description
ConnectionName	string	Filter	A descriptive name for the connection. Limit: 225 characters.
ContactId	reference	Filter Nillable	Contact associated with this connection. For information on IDs, see ID Field Type.
AccountId	reference	Filter Nillable	ID of the Account associated with this connection.
PrimaryContactId	reference	Filter	ID of the User associated with this connection.
ConnectionStatus	picklist	Filter	 Picklist of values. The picklist includes the following values: Sent Received Pending Accepted Rejected Inactive
ResponseDate	dateTime	Filter	The date and time that the connection was accepted or rejected.
CreatedDate	dateTime	Filter	The date and time that the connection was created.

Usage

Represents Salesforce to Salesforce connections. This object is referenced by lead, opportunity, account, contact, task, product, opportunity product, and custom objects that have been shared with other organizations, enabling you to determine which connections share the record with you. If the organization does not have Salesforce to Salesforce enabled, the PartnerNetworkConnection object is not available, and you cannot access it via the API.

PartnerNetworkRecordConnection

Represents a record shared between Salesforce organizations using Salesforce to Salesforce.

Supported Calls

Create(),Query(),Retrieve()

Field	Field Type	Field Properties	Description
ConnectionId	reference	Create	Required. ID of the connection a record is shared with.
		Filter	
		Nillable	
EndDate	dateTime	Filter	Date that sharing of the record was stopped.
		Nillable	
LocalRecordID	reference	Create	Required. ID of the shared record.
		Filter	
ParentRecordID	reference	Create	ID of the parent record of the shared record.
		Filter	
		Nillable	
PartnerRecordID	reference	Filter	ID of the shared record in the connection's
		Nillable	organization.
RelatedRecords	string	Create	A comma-separated list of API names for child records
		Filter	to be shared with a parent record.
		Nillable	
SendClosedTasks	boolean	Create	Forwards closed tasks related to the shared record.
		Defaulted on	
		Filter	
SendEmails	boolean	Create	Sends an email notifying the connection's representative
		Defaulted on create	recipients of a record will receive a notification email.
		Filter	
SendOpenTasks	boolean	Create	Forwards open tasks related to the shared record.
		Defaulted on create	
		Filter	
StartDate	dateTime	Filter	Date that the shared record was accepted.
		Nillable	
Status	picklist	Filter	The status of the shared record. One of the following values:
		Nillable	• Active (received)

Field	Field Type	Field Properties	Description
		Restricted picklist	 Active (sent) Connected Inactive Inactive (converted) Inactive (deleted) Danding (sent)
			• reliaing (selic)

When you Create() a PartnerNetworkRecordConnection, you forward a record to a connection. When you delete a PartnerNetworkRecordConnection, you stop sharing a record with a connection.

- To share a record, use the following fields: LocalRecordID and ConnectionId
- To share a child of a parent record, use the following fields: LocalRecordID, ConnectionId, and ParentRecordID
- To share a child of a parent record and its child records, use the following fields: LocalRecordID, ConnectionId, ParentRecordID, and RelatedRecords

If the organization does not have Salesforce to Salesforce enabled, the PartnerNetworkRecordConnection object is not available, and you cannot access it using the API.

Sample Code—Apex

The following example shows how to forward a record.

```
List<PartnerNetworkConnection)> connMap = new List<PartnerNetworkConnection>([select Id,
ConnectionStatus, ConnectionName from PartnerNetworkConnection where ConnectionStatus =
'Accepted']);
for(PartnerNetworkConnection network : connMap) {
    PartnerNetworkRecordConnection newrecord = new PartnerNetworkRecordConnection();
    newrecord.ConnectionId = network.Id;
    newrecord.LocalRecordId = accountId;
    newrecord.RelatedRecords = 'Contact,Opportunity,Orders_c';
    newrecord.SendClosedTasks = true;
    newrecord.SendOpenTasks = true;
    newrecord.SendEmails = true;
    insert newrecord;
}
```

The following example shows how to stop sharing a record.

```
List<PartnerNetworkRecordConnection> recordConns = new
List<PartnerNetworkRecordConnection>([select Id, Status, ConnectionId, LocalRecordId from
PartnerNetworkRecordConnection where LocalRecordId in :accounts]);
for(PartnerNetworkRecordConnection recordConn : recordConns) {
    if(recordConn.Status.equalsignorecase('Sent')){ //account is connected - outbound
```

```
delete nets;
```

}

PartnerRole

Represents a role for an account Partner, such as consultant, supplier, and so on. Corresponds to an SObject4 in which the ObjectType="PartnerRole".

Supported Calls

```
Query(),Retrieve(),CreateObject(),DescribeSObjects()
```

Special Access Rules

Customer Portal users cannot access this object.

Fields

Field	Field Type	Field Properties	Description
MasterLabel	string	Filter Nillable	Master label for this partner role value. This display value is the internal label that does not get translated. Maximum of 255 characters.
ReverseRole	picklist	Filter Nillable	Name of the reverse role that corresponds to this partner role. For example, if the role is "subcontractor," then the reverse role might be "general contractor." In the user interface, assigning a partner role to an account creates a reverse partner relationship so that both accounts list the other as a partner.
SortOrder	int	Filter Nillable	Number used to sort this value in the partner role picklist. These numbers are not guaranteed to be sequential, as some previous partner role values might have been deleted.

Usage

This object represents a value in the partner role picklist. The partner role picklist provides additional information about the role of a Partner, such as their corresponding reverse role. Your client application can invoke the Query() call on this object to retrieve the set of values in the partner role picklist, and then use that information while processing PartnerRole objects to determine more information about a given partner role. For example, the application could determine the reverse role of a given Partner Role value and the value of the ReverseRole property in the associated PartnerRole object.

This object is read-only via the API. With sufficient permissions, your client application can invoke the Query() and DescribeSObjects() calls on this object.

Period

Represents a fiscal period.

Supported Calls

Query(),Retrieve(),GetDeleted(),GetUpdated(),DescribeSObjects()

Special Access Rules

Customer Portal users cannot access this object.

Fields

Field	Field Type	Field Properties	Description
EndDate	date	Filter	The last date of the fiscal period.
IsForecastPeriod	boolean	Defaulted on create Filter	Indicates whether the period is associated with customizable forecasts (true) or not (false).
Number	int	Filter Nillable	If the labeling scheme of your fiscal year's quarters or months is numbered, this field indicates the relative number of the row.
PeriodLabel	picklist	Filter Nillable	If the months in your fiscal year use custom names, then this field contains the appropriate name for rows of type Month.
QuarterLabel	picklist	Filter Nillable	If the quarters in your fiscal year use custom names, then this field contains the appropriate name for rows of type Quarter.
StartDate	date	Filter	The first date of the fiscal period.
Туре	picklist	Filter	Indicates whether the period is of type Month, Quarter,
		Nillable	Week, or Year. Label is the field value.
		Restricted picklist	

Usage

This is a read-only object that is related to FiscalYearSettings.

Pricebook2

Represents a price book that contains the list of products that your organization sells.

Corresponds to an SObject4 in which the ObjectType="Pricebook2".



Note: In this release, price books are represented by Pricebook2 objects. The Pricebook object is no longer available for use.

Supported Calls

```
Create(),Update(),DescribeSObjects(),Query(),Retrieve(),CreateObject(),GetDeleted(),
GetUpdated()
```

Fields

Field	Field Type	Field Properties	Description
Description	string	Create	Text description of this object.
		Filter	
		Nillable	
		Update	
IsActive	boolean	Create	Indicates whether this object is active (true) or not
		Defaulted on create	(false). Inactive objects are hidden in many areas in the Salesforce user interface. You can change this field's value as often as necessary Label is Active
		Filter	
		Update	
IsDeleted	boolean	Defaulted on create	Indicates whether the record has been moved to the Recyc Bin (true) or not (false) Label is Deleted
		Filter	Dir (crue) of not (rarse). Laber is Deleted.
IsStandard	boolean	Defaulted on create	Indicates whether this object is the standard price book for the organization (true) or not (false). Every organization
		Filter	has one standard price book—all other price books are custom price books.
Name	string	Create	Required. Name of this object. This field is read-only for the
		Filter	standard price book. Label is Price Book Name .
		Update	

Usage

A price book is a list of products that your organization sells:

- Each organization has one standard price book that defines the standard or generic list price for each product or service that it sells.
- An organization can have multiple custom price books that can be used for specialized purposes, such as a discount price book, price books for different channels or markets, price books for select accounts or opportunities, and so on. While your client application can Create() and Update(), and custom price books, your client application can only Update() the standard price book.
- For some organizations, the standard price book might be the only price needed, but if you need to set up further price books, you can reference the standard price book when setting up list prices in custom price books.

Use this object to query standard and custom price books that have been configured for your organization. A common use of this object is to allow your client application to obtain valid Pricebook2 object IDs for use when configuring PricebookEntry records via the API.

Your client application can perform the following tasks on PricebookEntry objects:

- Query()
- Create() for the standard pricebook or custom pricebooks.
- Update()
- Change the IsActive field in Create () or Update () calls

PriceBook2, Product2, and PricebookEntry Relationships

In the API:

- Price books are represented by Pricebook2 objects (the Pricebook object has been deprecated and is no longer available).
- Products are represented by Product2 objects (the Product object has been deprecated and is not available as of version 8.0.).
- Each price book contains zero or more entries (represented by PricebookEntry records) that specify the products that are associated with the price book. A price book entry defines the price for which you sell a product at a particular currency.

These objects are defined only for those organizations that have products enabled as a feature. If the organization does not have the products feature enabled, the Pricebook2 object does not appear in the DescribeGlobal() call, and you cannot access it via the API.

If you delete a Pricebook2, while a line item references PricebookEntry in the price book, the line item is unaffected, but the Pricebook2 will be archived and unavailable from the API.

For a visual diagram of the relationships between Pricebook2 and other objects, see Product and Schedule Objects.

Price Book Setup

The process of setting up a price book via the API usually means:

- 1. Initially loading product data from your organization into Product2 objects (calling Create() for each product that you want to add).
- 2. For each Product2 object, creating a PricebookEntry that links the Product2 object to the standard Pricebook2. You need to define a standard price for a product at a given currency (if you have multicurrency enabled), before defining a price for that product in the same currency in a custom price book.
- 3. Creating a custom Pricebook2.
- 4. Querying the Pricebook2 object to obtain their IDs.
- 5. For each Pricebook2 object, creating a PricebookEntry for every Product2 that you want to add, specifying unique properties for each PricebookEntry (such as the UnitPrice and CurrencyIsoCode) as needed.

PricebookEntry

Represents a product entry (an association between a Pricebook2 and Product2) in a price book.

Corresponds to an SObject4 in which the ObjectType="PricebookEntry".

Supported Calls

```
Create(),Update(),Query(),Retrieve(),CreateObject(),GetDeleted(),GetUpdated(),
DescribeSObjects()
```

Field	Field Type	Field Properties	Description
CurrencyIsoCode	picklist	Filter Restricted picklist	Available only for organizations with the multicurrency feature enabled. Contains the ISO code for any currency allowed by the organization.
IsActive	boolean	Create Defaulted on create Filter Update	Indicates whether this object is active (true) or not (false). Although you can never delete PricebookEntry objects, your client application can set this flag to false. Inactive PricebookEntry objects are hidden in many areas in the Salesforce user interface. You can change this flag on a PricebookEntry object as often as necessary.
IsDeleted	boolean	Defaulted on create Filter	Indicates whether the record has been moved to the Recycle Bin (true) or not (false). Label is Deleted .
Name	string	Filter Nillable	Name of this PricebookEntry object. This read-only field references the value in the Name field of the Product2 object. Label is Product Name .
Pricebook2Id	reference	Create Filter	Required. ID of the Pricebook2 object with which this object is associated. Required field. This field must be specified in the Create() call. It cannot be changed in an Update() call. For information on IDs, see ID Field Type.
Product2Id	reference	Create Filter	ID of the Product2 object with which this object is associated. Required field. This field must be specified in the Create() call. It cannot be changed in an Update() call. For information on IDs, see ID Field Type.
ProductCode	string	Filter Nillable	Product code for this object. This read-only field references the value in the ProductCode field of the associated Product2 object.
UnitPrice	currency	Create Filter Update	Required. Unit price for this object. You can specify a value only if UseStandardPrice is set to false. Label is List Price.
UseStandardPrice	boolean	Create Defaulted on create Filter Update	Indicates whether this object uses the standard price defined in the standard Pricebook2 object (true) or not (false). If set to true, then the UnitPrice field is read-only, and the value will be the same as the UnitPrice value in the corresponding PricebookEntry in the standard price book (that is, the PricebookEntry object whose Pricebook2Id refers to the standard price book and whose Product2Id and CurrencyIsoCode are the same as this object). For PricebookEntry objects associated with the standard Pricebook2 object, this field must be set to true.

Use this object to define the association between your organization's products (Product2) and your organization's standard price book or to other, custom-defined price books (Pricebook2). Create one PricebookEntry record for each standard or custom price and currency combination for a product in a Pricebook2.

When your client application calls Create(), it must specify the IDs of the associated Pricebook2 object and Product2 object. Once created, your client application cannot update these IDs.

This object is defined only for those organizations that have products enabled as a feature. If the organization does not have the products feature enabled, then the PricebookEntry object does not appear in the DescribeGlobal() call, and you cannot access it.

If you delete a PricebookEntry while a line item references it, the line item is unaffected, but the PricebookEntry will be archived and unavailable from the API.

You must load the standard price for a product before you are permitted to load its custom price(s).

For a visual diagram of the relationships between PricebookEntry and other objects, see Product and Schedule Objects and PriceBook2, Product2, and PricebookEntry Relationships.

ProcessInstance

Represents an instance of a single, end-to-end approval process.

Supported Calls

Query(),Retrieve()

Special Access Rules

Customer Portal users cannot access this object.

Field	Field Type	Field Properties	Description
IsDeleted	boolean	Defaulted on create Filter	Indicates whether the record has been moved to the Recycle Bin (true) or not (false). Label is Deleted .
Status	picklist	Create Filter Restricted picklist Update	The status of this approval instance, for example Started, Pending, or Approved.
TargetObjectId	reference	Create Filter Update	ID of the object affected by this approval instance.

Use this object to create an approval process. See "What Are Approval Processes?" and "Managing Approval Processes" in the Salesforce online help.

ProcessInstanceHistory

This read-only object contains information about the current state of an approval process (ProcessInstance).

Supported Calls

DescribeGlobal(),DescribeSObject(),DescribeSObjects()

Special Access Rules

Customer Portal users cannot access this object.

Field	Field Type	Field Properties	Description
ActorId	reference	Filter	ID of the user who is currently assigned to this ProcessInstance.
Comments	string	Filter Nillable	Comments from any ProcessInstanceStep in the ProcessInstance.
IsDeleted	boolean	Defaulted on create Filter	Indicates whether the record has been moved to the Recycle Bin (true) or not (false). Label is Deleted .
IsPending	boolean	Defaulted on create Filter	Indicates whether the ProcessInstance is pending (true) or not (false).
OriginalActorId	reference	Filter	ID of the user who was originally assigned this ProcessInstance.
ProcessInstanceId	reference	Filter	ID of the ProcessInstance.
RemindersSent	int	Filter Nillable	Number of reminders that have been sent. Default is 0 (zero).
StepStatus	picklist	Filter Nillable Restricted picklist	Indicates the current status of the ProcessInstanceStep.

Field	Field Type	Field Properties	Description
TargetObjectId	reference	Filter Nillable	ID of the object being approved.

To retrieve this object, issue a describe call on an object, which will return a query result for each ProcessInstance since the object was created. You cannot use Query ().

This object respects field-level security on the parent object.

ProcessInstanceStep

Represents one step in an approval process (ProcessInstance).

Supported Calls

Query(),Retrieve()

Special Access Rules

Customer Portal users cannot access this object.

Field	Field Type	Field Properties	Description
ActorId	reference	Create Filter	ID of the user who is currently assigned to this approval step.
Comments	string	Create Filter Nillable	Limited to 4,000 bytes.
OriginalActorId	reference	Create Filter	ID of the user who was originally assigned to this approval step.
ProcessInstanceId	reference	Create Filter	ID of the ProcessInstance that this approval step belongs to.

Field	Field Type	Field Properties	Description
StepStatus	picklist	Create Filter Nillable Restricted picklist	The current status of this approval step, for example Pending or Approved. If the approval step requires unanimous approval and one approver rejects the request, the value of this field for the other approvers changes to NoResponse. Likewise, if approval is based on the first response and an approver responds, the value of this field for the other approvers changes to NoResponse.

Create, query, or retrieve a new step in an approval process (ProcessInstance).

ProcessInstanceWorkitem

Represents a user's pending approval request.

Supported Calls

Query(),Retrieve()

Special Access Rules

Customer Portal users cannot access this object.

Field	Field Type	Field Properties	Description
ActorId reference Create	Create	ID of the user who is currently responsible for approving an	
	Defaulted on create	Defaulted on create	approval request.
	Filter		
		Update	
IsDeleted	boolean	Defaulted on create	Indicates whether the record has been moved to the Recycle Bin (true) or not (false). Label is Deleted .
		Filter	
OriginalActorId	reference	Create	ID of the user who was originally assigned this approval
		Filter	request.
		Update	

Field	Field Type	Field Properties	Description
ProcessInstanceId	reference	Create Filter Update	ID of the ProcessInstance associated with this approval request.

Use this object to manage a pending approval request for a user.

Product2

Represents a product that your organization sells.

Corresponds to an SObject4 in which the ObjectType="Product2".



Supported Calls

```
Create(),Update(),Query(),Search(),Retrieve(),CreateObject(),GetDeleted(),GetUpdated(),
DescribeSObjects()
```

Field	Field Type	Field Properties	Description
CanUseQuantitySchedule	boolean	Defaulted on create Filter	Indicates whether the product can have a quantity schedule (true) or not (false). Label is Quantity Scheduling Enabled .
CanUseRevenueSchedule	boolean	Defaulted on create Filter	Indicates whether the product can have a revenue schedule (true) or not (false). Label is Revenue Scheduling Enabled .
ConnectionReceivedID	reference	Filter Nillable	ID of the PartnerNetworkConnection that shared this record with your organization. This field is only available if you have enabled Salesforce to Salesforce.
ConnectionSentID	reference	Filter Nillable	ID of the PartnerNetworkConnection that you shared this record with. This field is only available if you have enabled Salesforce to Salesforce. Beginning with API version 15.0, the ConnectionSentID field is no longer supported. The ConnectionSentID

Field	Field Type	Field Properties	Description
			field is still be visible, but the value is null. You can use the new PartnerNetworkRecordConnection object to forward records to connections.
CurrencyIsoCode	picklist	Filter Restricted picklist	Available only for organizations with the multicurrency feature enabled. Contains the ISO code for any currency allowed by the organization.
DefaultPrice	currency	Create	The default price for this record.
		Filter	
		Update	
Description	textarea	Create	A text description of this record. Label is
		Filter	Product Description.
		Nillable	
		Update	
Family	picklist	Filter Nillable	Name of the product family associated with this record. Product families are configured as picklists in the Salesforce user interface. To obtain a list of valid values, call DescribeSObjects() and process the DescribeSObjectResult for the values associated with the Family field. Label is Product Family .
IsActive	boolean	Create Defaulted on create Filter Update	Indicates whether this record is active (true) or not(false). Inactive Product2s are hidden in many areas in the Salesforce user interface. You can change the ISActive flag on a Product2 object as often as necessary. Label is Active .
IsDeleted	boolean	Defaulted on create Filter	Indicates whether the record has been moved to the Recycle Bin (true) or not (false). Label is Deleted .
Name	string	Create Filter Update	Required. Default name of this record. Label is Product Name .
NumberOfQuantityInstallments	int	Filter Nillable	If the product has a quantity schedule, the number of installments.

Field	Field Type	Field Properties	Description
NumberofRevenueInstallments	int	Filter	If the product has a revenue schedule, the
		Nillable	number of installments.
ProductCode	string	Create	Default product code for this record. The
		Filter	product code naming pattern is defined by
		Nillable	your organization.
		Update	
QuantityInstallmentPeriod	picklist	Filter	If the product has a quantity schedule, the
		Nillable	amount of time covered by the schedule.
		Restricted picklist	
QuantityScheduleType	picklist	Filter	The type of the quantity schedule, if the
		Nillable	product has one.
		Restricted picklist	
RevenueInstallmentPeriod	picklist	Filter	If the product has a revenue schedule, the
		Nillable	period of time covered by the schedule.
		Restricted picklist	
RevenueScheduleType	picklist	Filter	The type of the revenue schedule, if the
		Nillable	product has one.
		Restricted picklist	
ConnectionReceivedID	reference	Filter	ID of the PartnerNetworkConnection that
		Nillable	shared this record with your organization. This field is only available if you have enabled Salesforce to Salesforce.
ConnectionSentID	reference	Filter	ID of the PartnerNetworkConnection that
		Nillable	you shared this record with. This field is only available if you have enabled Salesforce to Salesforce. Beginning with API version 15.0, the ConnectionSentID field is no longer supported. The ConnectionSentID field is still be visible, but the value is null. You can use the new PartnerNetworkRecordConnection object to forward records to connections.

Schedule Fields

This object has several fields that are only used for schedules (for example, annuities). The API supports quantity and revenue schedules on this object. Schedules are available only for those organizations that have the products and schedules features enabled. If the organization does not have the schedules feature, the schedule fields do not appear in the DescribeSObjectResult, and you cannot Query(), Create(), or Update() the fields.

Schedule Enabled Flags

When enabling the schedules feature, organizations can decide whether to enable quantity schedules, revenue schedules, or both. In addition, you can use the API to control quantity and revenue scheduling at the product level via the CanUseQuantitySchedule and CanUseRevenueSchedule flags. A value of true for either flag indicates that the product and any OpportunityLineItems can have a schedule of that type. These flags can be set via a Create() or Update() call.

Default Schedule Fields

The remaining schedule fields for this object define default schedules. Default schedule values are used to create an OpportunityLineItemSchedule when an OpportunityLineItem is created for the Product.

Field	Valid Values
RevenueScheduleType	Divide, Repeat
RevenueInstallmentPeriod	Daily, Weekly, Monthly, Quarterly, Yearly
NumberOfRevenueInstallments	Integer between 1 to 150, inclusive.
QuantityScheduleType	Divide, Repeat
QuantityInstallmentPeriod	Daily, Weekly, Monthly, Quarterly, Yearly
NumberOfQuantityInstallments	Integer between 1 to 150, inclusive.

The default schedule fields support the following valid values (all fields are also nillable).

When you attempt to set the schedule fields via a Create () or Update () call, the API applies cross-field integrity checks. The integrity requirements are:

- If the schedule type is nil, the installment period and number of installments must be nil.
- If the schedule type is set to any value, then the installment period and number of installments must be non-nil.

Any Create () or Update () calls that fail these integrity checks are rejected with an error.

These default schedule fields, as well as CanUseQuantitySchedule and CanUseRevenueSchedule, are restricted picklist fields and are available only if the organization has the schedules feature enabled.

Usage

Use this object to define the default product information for your organization. This object is associated by reference with Pricebook2 objects via PricebookEntry objects. The same product can be represented in different price books as price book entries. In fact, the same product can be represented multiple times (as separate PricebookEntry records) in the same price book with different prices and/or currencies. A product can only have one price for a given currency within the same price book. To be used in custom price books, all standard prices must be added as price book entries to the standard price book.

You can query the products that have been configured for your organization. For example, you can allow your client application to obtain valid product IDs for use when configuring PricebookEntry records via the API. Your client application can perform the following tasks on PricebookEntry objects:

• Query()

- Create() for the standard pricebook or custom pricebooks.
- Update()
- Change the IsActive field in Create () or Update () calls

This object is defined only for those organizations that have products enabled as a feature. If the organization does not have the products feature, this object does not appear in the DescribeGlobal() call, and you cannot use DescribeSObjects() or Query() with this object.

If you try to delete a product via the API but there is an opportunity that uses that product, the delete fails. The workaround is to delete the product in the Salesforce user interface, which gives you an option to archive the product.

For a visual diagram of the relationships between Product2 and other objects, see Product and Schedule Objects and PriceBook2, Product2, and PricebookEntry Relationships.



Note: With Spring '09, workflow rules, validation rules and Apex triggers on opportunities and opportunity products will fire when an update to a child opportunity product or schedule causes an update to the parent record. This means your custom application logic will be enforced when there are updates to the parent record, ensuring higher data quality and compliance with your organization's business policies.

By default, all organizations signed up after the Spring '09 release will see this behavior. Starting in March 09, salesforce.com will begin rolling out the New Opportunity Save Behavior update in phases. Once available, you will see the update on the Critical Updates page. The update will be auto-enabled for organizations not currently using any of the customizations listed below. These customers will be required to review the update and take the necessary actions in order to enable the update prior to its auto-activation date. This change is effective in all versions of the API once the New Opportunity Save Behavior update is activated for an organization.

For more information, see What is the Spring '09 New Opportunity Save Behavior Update? in the Salesforce online help.

Profile

Represents a profile, which defines a set of permissions to perform different operations, such as querying, adding, updating, or deleting information.

Corresponds to an SObject4 in which the ObjectType="Profile".

See "Managing Profiles" and "User Permissions on Profiles" in the Salesforce online help.

Supported Calls

Update(),Query(),Retrieve(),CreateObject(),GetDeleted(),GetUpdated(),DescribeSObjects()

Special Access Rules

Customer Portal users cannot access this object.

Field	Field Type	Field Properties	Valid Values
Name	string	Filter	The name of the profile.
		Update	

Field	Field Type	Field Properties	Valid Values
Permission <i>PermissionName</i>	boolean	Update	One field for each permission. If true, users assigned to this profile have the named permission. The number of fields varies depending on the permissions for the organization and license type.
UserLicenseID	ID	Filter Nillable	ID of the UserLicense associated with this profile.
UserType	picklist	Nillable Filter Nillable Restricted picklist	 The category of user license. Each UserType is associated with one or more UserLicense records. Each UserLicense is associated with one or more profiles. In API version 10.0 and later, valid values include: Standard: Salesforce user license. This user type also includes Salesforce Platform and Salesforce Platform One user licenses. Label is Standard. PowerPartner: PRM user whose access is limited because he or she is a partner and typically accesses the application through a partner portal. Label is Partner. CustomerSuccess: user whose access is limited because he or she is an organization's customer and accesses the application through a customer portal. Label is Customer Portal User. PowerCustomerSuccess: user whose access is limited because he or she is an organization's customer and accesses the application through a customer portal. Label is Customer Portal User. PowerCustomerSuccess: user whose access is limited because he or she is an organization's customer and accesses the application through a customer portal. Label is Customer Portal Manager. Users with this license type can view and edit data they directly own or data owned by or shared with users below them in the customer portal role hierarchy. UserType replaces LicenseType, which was deprecated in API version 10.0. In API versions 8.0 and 9.0 LicenseType is still available with the following valid values: AUL: Force.com user license. Label is Salesforce. Ault: Force.com user license. Label is Salesforce. Salesforce: Salesforce user license. Label is Salesforce. PackageManager: user who can create and work with managed packages for Force.com AppExchange. Label is a partner and typically accesses the application through a partner portal. Label is Partner.
			• CustomerUser: user whose access is limited because he or she is an organization's customer and accesses the application through a customer portal. Label is Customer Portal User .

Field	Field Type	Field Properties	Valid Values
			• CustomerManager: user whose access is limited because he or she is an organization's customer and accesses the application through a customer portal. Label is Customer Portal Manager .
			Users with this license type can view and edit data they directly own or data owned by or shared with users below them in the customer portal role hierarchy.

Use the Profile object to Query() the set of currently configured user profiles in your organization. Your client application can use Profile objects to obtain valid profile IDs for use when querying or modifying users through the API. Your client application can Query() and Update() Profiles.

In the Salesforce user interface, profiles can be used to assign user licenses from specific pools (Force.com Platform user license or Salesforce user license, for example). If a user is assigned to a profile with a different license type, the number of available licenses in the old license type pool increases, one per user changed, and decreases by the same amount in the new license type pool. For more information, see the Salesforce online help.

QuantityForecast

Represents a quantity-based forecast.

The API also provides revenue-based forecasts; see RevenueForecast.

Supported Calls

Create(),Query(),Retrieve(),GetDeleted(),GetUpdated(),DescribeSObjects()

Special Access Rules

- Requires the "View All Data" permission.
- Customer Portal users cannot access this object.

Field	Field Type	Field Properties	Description
Closed	double	Filter Nillable	Read-only. A rollup of opportunities or opportunity line items that have closed in this period.
Commit	currency	Filter Nillable	The owner's Commit total.

Field	Field Type	Field Properties	Description
CommitComment	string	Filter Nillable Update	Read-only. The comment entered when the owner edited his or her Commit total from the Adjusted Total link on the forecast edit page.
CommitOverride	double	Filter Nillable Update	Read-only. The owner's override of their own My Commit total.
DefaultRollupCommit	double	Filter Nillable	Read-only. The owner's standard Commit rollup, including their own opportunities and forecast-level overrides from subordinate users in the role hierarchy.
DefaultRollupUpside	double	Filter Nillable	Read-only. The owner's standard Best Case rollup, including their own opportunities and forecast-level overrides from subordinate users in the role hierarchy.
InvalidationDate	dateTime	Filter Nillable	Read-only. If not blank, indicates that the rollup numbers for Forecast Override fields that represent calculated (summarized) amounts may not be up to date.
ManagerChoiceCommit	picklist	Filter Nillable Restricted picklist Update	 Read-only. The manager's choice regarding the commit amount: DefaultRollup: Use the manager's default commit rollup for owner's forecast, which reflects the manager's opportunity forecast overrides. AcceptForecast (default selection): Accept the forecast owner's Adjusted Total commit amount, which may or may not be an override. ManagerManualOverride: Use the manager's manual override. OpportunityOnlyRollup: Use the opportunity rollup, including opportunity forecast overrides, but excluding any forecast-level (Adjusted Total) overrides.
ManagerChoiceUpside	picklist	Filter Nillable Restricted picklist Update	 Read-only. The manager's choice regarding the best case amount: DefaultRollup: Use the manager's default best case rollup for owner's forecast, which reflects the manager's opportunity forecast overrides. AcceptForecast (default selection): Accept the forecast owner's Adjusted Total best case amount, which may or may not be an override. ManagerManualOverride: Use the manager's manual override. OpportunityOnlyRollup: Use the opportunity rollup, including opportunity forecast overrides, but

Field	Field Type	Field Properties	Description
			excluding any forecast-level (Adjusted Total) overrides.
ManagerClosed	double	Filter	Read-only. The manager's closed total for the owner's
		Nillable	forecast, including any opportunity or opportunity product overrides made by the manager.
ManagerCommit	currency	Filter	The manager's Commit total.
		Nillable	
ManagerCommitOverride	double	Filter	Read-only. The manager's manual override of the
		Nillable	forecast owner's Commit total. Represents an option in the override pop-up window, which allows managers
		Update	to choose how to roll up the forecast numbers of a direct report.
ManagerDefaultRollupCommit	double	Filter	Read-only. The manager's standard Commit rollup
		Nillable for the forecast owner. Represer override pop-up window, which choose how to roll up the foreca report.	for the forecast owner. Represents an option in the override pop-up window, which allows managers to choose how to roll up the forecast numbers of a direct report.
ManagerDefaultRollupUpside	double	Filter	Read-only. The manager's standard Best Case rollup
		Nillable	for the forecast owner. Represents an option in the override pop-up window, which allows managers to choose how to roll up the forecast numbers of a direct report.
ManagerId	reference	Filter	Read-only. ID of the direct manager of the user who
		Nillable	owns this forecast. For information on IDs, see ID Field Type.
ManagerOpportunityRollupCommit	ortunityRollupCommit double Filter Read- Nillable opport overric windo up the	Filter	Read-only. The manager's view of the forecast owner's
		opportunity-level Commit rollup, ignoring all forecast overrides. Represents an option in the override pop-up window, which allows managers to choose how to roll up the forecast numbers of a direct report.	
ManagerOpportunityRollupUpside	double	Filter	Read-only. The manager's view of the forecast owner's
Nillable o o w	opportunity-level Best Case rollup, ignoring all forecast overrides. Represents an option in the override pop-up window, which allows managers to choose how to roll up the forecast numbers of a direct report.		
ManagerPipeline	double	Filter	The manager's pipeline total for the owner's forecast,
		Nillable	including any opportunity or opportunity product overrides made by the manager.
ManagerUpside	currency	Filter	The manager's Best Case total.
		Nillable	
ManagerUpsideOverride	double	Filter	The manager's manual override of the forecast owner's
		Nillable	Best Case total. Represents an option in the override

Field	Field Type	Field Properties	Description
		Update	pop-up window, which allows managers to choose how to roll up the forecast numbers of a direct report.
OpportunityRollupClosed	double	Filter	The owner's Closed total for his or her opportunities
		Nillable	only.
OpportunityRollupCommit	double	Filter	The owner's Commit total for his or her opportunities
		Nillable	only.
OpportunityRollupPipeline	double	Filter	The owner's Pipeline total for his or her opportunities
		Nillable	only.
OpportunityRollupUpside	double	Filter	The owner's Best Case total for his or her opportunities
		Nillable	only.
OwnerId	reference	Create	ID of the User who owns this forecast. Required on
		Defaulted on create	create.
		Filter	
PeriodId	reference	Filter	The ID of the Period that contains the StartDate.
		Nillable	
Pipeline	double	Filter	The total pipeline rollup from subordinates in the role
		Nillable	hierarchy, including the owner's opportunities.
ProductFamily	picklist	Create	The value chosen in the Product Family picklist, which
		Filter	 Can be configured at Setup ➤ Customize ➤ Products ➤ Fields. This field is relevant if you have chosen "Use
		Nillable	Product Families" as the Forecast Type at Setup ➤ Customize ➤ Forecast ➤ Settings. If you are not forecasting by product family or if the forecast represents opportunities that are not associated with a product family, then this field is blank. Required on create.
Quota	double	Create	The quota amount for the period. You can Create () and Update (). Requires the "Modify All Data" and
		Filter	"Manage Users" permission. Required on create.
		Nillable	
		Update	
StartDate	date	Create	The start date of this forecast. The ID of the Period that contains this date is written to the Period I dield
		Filter	if it changes. A new Period is created if none exists.
		Nillable	Required on create.

Field	Field Type	Field Properties	Description
Upside	currency	Filter Nillable	The owner's Best Case total.
UpsideComment	string	Filter Nillable Update	The comment entered when the owner edited his or her Best Case total.
UpsideOverride	double	Filter Nillable Update	Read only. The owner's override of their own My Best Case total.

Query this object to support customizable forecasts based on quantities.

You can update one field, Quota, which means that you can mass update sales users' quotas instead of updating them one by one in the Salesforce user interface. Editing Quota, which means that you can mass update sales users' quotas instead of updating them one by one in the Salesforce user interface. Editing Quota requires the "Modify All Data" and "Manage Users" permissions.

The rollup fields always reflect opportunity and opportunity product overrides by the forecast owner or one of the forecast owner's subordinates in the role hierarchy. In addition, the manager rollup fields include overrides by the forecast owner's direct manager in the role hierarchy.

Some of the rollup fields ignore forecast-level (**Adjusted Total**) overrides, but they never ignore opportunity forecast overrides that are visible to the owner or manager.

QuantityForecastHistory

Represents historical information about quantity-based forecasts that have been submitted (saved) in the Salesforce user interface.

Supported Calls

Query(),Retrieve(),GetDeleted(),GetUpdated()DescribeSObjects(),

Special Access Rules

- Requires the "View All Data" permission.
- Customer Portal users cannot access this object.

Fields

Field	Field Type	Field Properties	Description
Closed	currency	Filter	The closed amount of the forecast.
		Nillable	
Commit	currency	Filter	The commit amount of the forecast.
		Nillable	
CommitComments	string	Filter	Comments about the commit value.
		Nillable	
CommitOverridden	boolean	Defaulted on create	Indicates whether the commit value was overridden (true) or not (false).
		Filter	
CurrencyIsoCode	picklist	Filter	Available only for organizations with the multicurrency
		Restricted picklist	feature enabled. Contains the ISO code for any currency allowed by the organization.
ForecastOverrideId	reference	Filter	ID of the related forecast override. For information on IDs, see ID Field Type. Label is Revenue Forecast ID .
Pipeline	currency	Filter	The pipeline amount of the forecast.
		Nillable	
Quota	currency	Filter	The quota amount of the forecast.
		Nillable	
Upside	currency	Filter	The best case amount of the forecast.
		Nillable	
UpsideComments	string	Filter	Comments about the upside value.
		Nillable	
UpsideOverridden	boolean	Defaulted on create	Indicates whether the upside value was overridden (true) or not (false).
		Filter	

Usage

This is a read-only object specific to customizable forecasting.

When a user submits a revenue-based forecast in the Salesforce user interface, a new record is created. If the same forecast is ever resubmitted, additional records are added. The CreatedDate of a record reflects the day on which the forecast was submitted. For more information about customizable forecasts, see the Salesforce online help. This object respects field level security on the parent object.

QueueSobject

Represents the mapping between a queue Group and the SObject4 types associated with the queue, including custom objects.

Supported Calls

Create(),DescribeSObjects(),Query(),Retrieve(),GetDeleted(),GetUpdated()

Special Access Rules

Customer Portal users cannot access this object.

Fields

Field	Field Type	Field Properties	Description
QueueId	reference	Create Filter	The ID of a queue. For information on IDs, see ID Field Type.
SobjectType	picklist	Create Filter	A list of object types that can be associated with the queue specified by the QueueId.

Usage

Use this object to associate a queue with the SObject4 that can be associated with the queue, including custom objects.

A queue is a Group whose Type is Queue.

RecordType

Represents a record type. See "Managing Record Types" in the Salesforce online help.

Corresponds to an SObject4 in which the ObjectType="RecordType".

Supported Calls

```
Create(),Update(),Query(),Retrieve(),CreateObject(),GetDeleted(),GetUpdated(),
DescribeSObjects()
```

Special Access Rules

Customer Portal users cannot access this object.

Field	Field Type	Field Properties	Description
BusinessProcessId	reference	Create	Required. ID of an associated BusinessProcess. For information on IDs. see ID Field Type
		Filter	monnation on 123, see 12 1 feft Type.
		Nillable	
		Update	
Description	string	Create	A description of this record. Limit: 255 characters.
		Filter	
		Nillable	
		Update	
DeveloperName	string	Create	Required. The unique name of the object in the API. The
		Filter	name can contain only alphanumeric characters and must
		Nillable	naming conflicts on package installations. With this field, a
		Update	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 .
IsActive boolean	boolean I C	Defaulted	Indicates whether this record is active (true) or not
		on create	(false). Only active record types can be applied to records. Label is Active .
		Filter	
		Update	
IsPersonType	boolean	Defaulted on create	Indicates whether this record has been designated as a PersonAccount (true) or not (false). Visible only if the
		Filter	organization has the person account feature enabled. For more information about person accounts, see Person Account Record Types, and the Salesforce online help.
Name	string	Create	Required. Label of the record type in the Salesforce user
		Filter	interface. Limit: 80 characters. Label is Record Type Label.
		Update	
NamespacePrefix	string	Create	The namespace prefix assigned to this object when it was
		Filter	created as part of a managed package. Null if this object is
	Ni	Nillable	more information about managed packages and namespace prefixes, see "About Managed Packages" in the Salesforce online help.
			This field cannot be accessed unless the logged-in user has the "Customize Application" permission.
SobjectType picklist Create		Create	Object to which this record type applies, including custom
Filter	Filter	objects.	

Field	Field Type	Field Properties	Description
		Restricted picklist	

Use this object to offer different BusinessProcesses and subsets of picklist values to different users based on their Profile. Your client application can invoke the DescribeSObjects() and Query() calls on RecordType objects.

The following objects have a RecordTypeId field: Account, Campaign, Case, Contact, Contract, Lead, Opportunity, and Solution. Also, custom objects are supported. Client applications can set this field in Create() or Update() calls on these objects, specifying a valid record type ID associated with these objects. A client application can retrieve the list of valid record type IDs for a given object by calling Query() on the RecordType object. For more information, see RecordTypeId.

RevenueForecast

Represents a revenue-based forecast.

The API also provides quantity-based forecasts; see QuantityForecast.

Supported Calls

```
Create(),Create(),Query(),Retrieve(),Update(),GetDeleted(),GetUpdated(),
DescribeSObjects()
```

Special Access Rules

Customer Portal users cannot access this object.

Field	FieldType	Field Properties	Description
Closed	currency	Create Filter	Read only. A rollup of opportunities or opportunity line items that have closed in this period.
Commit	currency	Filter Nillable	The owner's Commit total.
CommitComment	string	Filter Nillable Update	Read only. The comment entered when the owner edited his or her Commit total from the Adjusted Total link on the forecast edit page.
CommitOverride	currency	Filter Nillable Update	Read only. The owner's override of their own My Commit total.

Field	FieldType	Field Properties	Description
CurrencyIsoCode	picklist	Filter Restricted picklist Update	Available only for organizations with the multicurrency feature enabled. Contains the ISO code for any currency allowed by the organization. You can use Update () on this field and Quota only.
DefaultRollupCommit	currency	Filter Nillable	Read only. The owner's standard Commit rollup, including their own opportunities and forecast-level overrides from subordinate users in the role hierarchy.
DefaultRollupUpside	currency	Filter Nillable	Read only. The owner's standard Best Case rollup, including their own opportunities and forecast-level overrides from subordinate users in the role hierarchy.
InvalidationDate	dateTime	Filter Nillable	Read only. If not blank, indicates that the rollup numbers for Forecast Override fields that represent calculated (summarized) amounts may not be up to date.
ManagerChoiceCommit	picklist	Filter Nillable Restricted picklist Update	 Read only. The manager's choice regarding the commit amount: DefaultRollup: Use the manager's default commit rollup for owner's forecast, which reflects the manager's opportunity forecast overrides. AcceptForecast (default selection): Accept the forecast owner's Adjusted Total commit amount, which may or may not be an override. ManagerManualOverride: Use the manager's manual override. OpportunityOnlyRollup: Use the opportunity rollup, including opportunity forecast overrides, but excluding any forecast-level (Adjusted Total) overrides.
ManagerChoiceUpside	picklist	Filter Nillable Restricted picklist Update	 Read only. The manager's choice regarding the best case amount: DefaultRollup: Use the manager's default best case rollup for owner's forecast, which reflects the manager's opportunity forecast overrides. AcceptForecast (default selection): Accept the forecast owner's Adjusted Total best case amount, which may or may not be an override. ManagerManualOverride: Use the manager's manual override. OpportunityOnlyRollup: Use the opportunity rollup, including opportunity forecast overrides, but excluding any forecast-level (Adjusted Total) overrides.

Field	FieldType	Field Properties	Description
ManagerClosed	currency	Filter Nillable	Read only. The manager's closed total for the owner's forecast, including any opportunity or opportunity product overrides made by the manager.
ManagerCommit	currency	Filter	The manager's Commit total.
		Nillable	
ManagerCommitOverride	currency	Filter	Read only. The manager's manual override of the forecast owner's Commit total Represents an option
		Nillable	in the override popup window, which allows
		Update	managers to choose how to roll up the forecast numbers of a direct report.
ManagerDefaultRollupCommit	currency	Filter	Read only. The manager's standard Commit rollup
		Nillable	override pop-up window, which allows managers to choose how to roll up the forecast numbers of a direct report.
ManagerDefaultRollupUpside	currency	Filter Nillable	Read only. The manager's standard Best Case rollup for the forecast owner. Represents an option in the override pop-up window, which allows managers to choose how to roll up the forecast numbers of a direct report.
ManagerId	reference	Filter Nillable	Read only. ID of the direct manager of the user who owns this forecast. For information on IDs, see ID Field Type.
ManagerOpportunityRollupCommit	currency	Filter Nillable	The manager's view of the forecast owner's opportunity-level Commit rollup, ignoring all forecast overrides. Represents an option in the override pop-up window, which allows managers to choose how to roll up the forecast numbers of a direct report.
ManagerOpportunityRollupUpside	currency	Filter Nillable	The manager's view of the forecast owner's opportunity-level Best Case rollup, ignoring all forecast overrides. Represents an option in the override pop-up window, which allows managers to choose how to roll up the forecast numbers of a direct report
ManagerPipeline	currency	Filter Nillable	The manager's pipeline total for the owner's forecast, including any opportunity or opportunity product overrides made by the manager.
ManagerTerritoryId	reference	Filter	ID of the manager's UserRole or Territory.
		Nillable	
ManagerUpside	currency	Filter	The manager's Best Case total.
		Nillable	

Field	FieldType	Field Properties	Description
ManagerUpsideOverride	currency	Filter Nillable Update	The manager's manual override of the forecast owner's Best Case total. Represents an option in the override pop-up window, which allows managers to choose how to roll up the forecast numbers of a direct report.
OpportunityRollupClosed	currency	Filter Nillable	The owner's Closed total for his or her opportunities only.
OpportunityRollupCommit	currency	Filter Nillable	The owner's Commit total for his or her opportunities only.
OpportunityRollupPipeline	currency	Filter Nillable	The owner's Pipeline total for his or her opportunities only.
OpportunityRollupUpside	currency	Filter Nillable	Read only. The owner's Best Case total for his or her opportunities only.
OwnerId	reference	Create Defaulted on create Filter	ID of the User who owns this forecast. Required on create.
PeriodId	reference	Filter Nillable	The ID of the Period that contains the StartDate.
Pipeline	currency	Filter Nillable	The total pipeline rollup from subordinates in the role hierarchy, including the owner's opportunities.
ProductFamily	picklist	Create Filter Nillable	The value chosen in the Product Family picklist, which can be configured at Setup $>$ Customize $>$ Products $>$ Fields . This field is relevant if you have chosen "Use Product Families" as the Forecast Type at Setup $>$ Customize $>$ Forecast $>$ Settings . If you are not forecasting by product family or if the forecast represents opportunities that are not associated with a product family, then this field is blank. Otherwise, this field is required on create.
Quota	currency	Create Filter Nillable Update	The quota amount for the period. You can use Update() with this field and CurrencyIsoCode, and for Quota only, Create(). Requires the "Modify All Data" and "Manage Users" permission. Required on create.
StartDate	date	Create Filter	The start date of this forecast. The period ID of the period that contains this date is written to the

Field	FieldType	Field Properties	Description
		Nillable	PeriodId field if it changes. A new Period is created if none exists. Required on create.
TerritoryId	reference	Create Filter Nillable	ID of the forecast owner's UserRole or Territory. Required on create if Territory Management is enabled (if this field is available).
Upside	currency	Filter Nillable	The owner's Best Case total.
UpsideComment	string	Filter Nillable Update	Read only. The comment entered when the owner edited his or her Best Case total. Label is Best Case Comment .
UpsideOverride	currency	Filter Nillable Update	Read only. The owner's override of their My Best Case total. Label is Best Case Override .

Query this object to support customizable forecasts based on revenue. Requires the "View All Data" permission.

You can also update CurrencyIsoCode and Quota, which means that you can mass update sales users' quotas instead of updating them one by one in the Salesforce user interface. Editing Quota requires "Modify All Data" and "Manage Users" permissions.

The rollup fields always reflect opportunity and opportunity product overrides by the forecast owner or one of the forecast owner's subordinates in the role hierarchy. In addition, the manager rollup fields include overrides by the forecast owner's direct manager in the role hierarchy.

Some of the rollup fields ignore forecast-level (**Adjusted Total**) overrides, but they never ignore opportunity forecast overrides that are visible to the owner or manager.

RevenueForecastHistory

Represents historical information about revenue-based forecasts that have been submitted (saved) in the Salesforce user interface.

Supported Calls

Query(),Retrieve(),GetDeleted(),GetUpdated(),DescribeSObjects()

Special Access Rules

- Requires the "View All Data" permission.
- Customer Portal users cannot access this object.

Fields

Field	Field Type	Field Properties	Description
Closed	currency	Filter	The closed amount of the forecast.
		Nillable	
Commit	currency	Filter	The commit amount of the forecast.
		Nillable	
CommitComments	string	Filter	Comments about the commit value.
		Nillable	
CommitOverridden	boolean	Defaulted on create	Indicates whether the commit value was overridden (true) or not (false).
		Filter	
CurrencyIsoCode	picklist	Filter	Available only for organizations with the multicurrency
		Restricted picklist	feature enabled. Contains the ISO code for any currency allowed by the organization.
ForecastOverrideId	reference	Filter	ID of the related forecast override. For information on IDs, see ID Field Type. Label is Revenue Forecast ID .
Pipeline	currency	Filter	The pipeline amount of the forecast.
		Nillable	
Quota	currency	Filter	The quota amount of the forecast.
		Nillable	
Upside	currency	Filter	The best case amount of the forecast.
		Nillable	
UpsideComments	string	Filter	Comments about the upside value.
		Nillable	
UpsideOverridden	boolean	Defaulted on create	Indicates whether the upside value was overridden (true) or not (false).
		Filter	

Usage

This is a read-only object specific to customizable forecasting.

When a user submits a revenue-based forecast in the Salesforce user interface, a new record is created. If the same forecast is ever resubmitted, additional records are added. The CreatedDate of a record reflects the day on which the forecast was submitted. For more information about customizable forecasts, see the Salesforce online help. This object respects field-level security on the parent object.

Scontrol

Note: Salesforce expects to deprecate s-controls, including the Scontrol object, in the near future. We recommend that you move your s-controls to Visualforce where possible. We will continue to support s-controls and the Scontrol object for the time being. For more information about our support and deprecation policy, see Office Toolkit Support Policy on page 11.

Represents a custom s-control, which is custom content that is hosted by the system but executed by client applications. An s-control can contain any type of content that you can display or run in a Web browser. See "Defining Custom S-Controls" in the Salesforce online help.

Corresponds to an SObject4 in which the ObjectType="Scontrol".

Supported Calls

```
Create(),Update(),DescribeSObjects(),Query(),Retrieve(),CreateObject(),GetDeleted(),
GetUpdated()
```

Special Access Rules

- Your organization must be using Enterprise, Developer, or Unlimited Edition and be enabled for custom s-controls.
- · Customer Portal users cannot access this object.

Field	Field Type	Field Properties	Description
Binary	base64	Create	Binary content of this custom s-control, such as an ActiveX control or a Java archive. Can be specified when your client application calls Create(), but not when calling Update(). Limit: 5 MB.
		Nillable	
		Update	
BodyLength	int	Filter	The length of the custom s-control. Label is Binary Length .
ContentSource	picklist	Create	Specify the source of the s-control content, either custom
		Filter	HTML, a snippet (s-controls that are designed to be included in other s-controls), or a URL.
		Nillable	
		Restricted picklist	
		Update	
Description	string	Create	Description of the custom s-control.
		Filter	
		Nillable	
		Update	
DeveloperName	string	Create	The unique name of the object in the API. The name can contain only alphanumeric characters and must begin with
		Filter	

Field	Field Type	Field Properties	Description
		Nillable	a letter. In managed packages, this field prevents naming
		Update	conflicts on package installations. With this field, a develop can change the object's name in a managed package and the changes are reflected in a subscriber's organization. Label S-Control Name.
EncodingKey	picklist	Create	Picklist of character set encodings, including ISO-08859-1 UTF-8, EUC, JIS, Shift-JIS, Korean (ks_c_5601-1987), Simplified Chinese (GB2312), and Traditional Chinese (Big5).
		Filter	
		Restricted picklist	
		Update	
Filename	string	Create	An uploaded object to display when the custom s-control is
		Filter	added to a custom link. Can be a Java applet, an ActiveX
		Nillable	control, of any other type of desired content.
		Update	
HtmlWrapper	textarea	Create	Required. HTML page that will be delivered when the user
		Update	views this custom s-control. This HTML page can be the entire content of the custom s-control, or it can reference the binary. Limit: 1,048,576 characters. Label is HTML Bod
Name	string	Create	Required. Name of this custom s-control. Label is Label
		Filter	
		Nillable	
		Update	
NamespacePrefix	string	Create	The namespace prefix assigned to this object when it was
		Filter	created as part of a managed package. Null if this object is not part of a managed package. Limit is 15 characters. For
		Nillable	more information about managed packages and namespa prefixes, see "About Managed Packages" in the Salesforce online help.
			You can use this field to identify rows that belong to a particular managed package, for example, identifying the tabset that was installed with a particular managed package.
SupportsCaching	boolean	Create	Indicates whether the s-control supports caching (true) or not (false).
		Filter	
		Update	

Use custom s-controls to manage custom content that extends the base Salesforce application functionality. All users can view custom s-controls, but the "Customize Application" permission is required to Create() or Update() custom s-controls.

ScontrolLocalization

When the translation workbench is enabled for your organization, provides the translation of the field label of an s-control. For information on the translation workbench, see the Salesforce online help.

Supported Calls

Create(),Update(),DescribeSObjects(),Query(),Retrieve(),CreateObject(),GetDeleted(), GetUpdated()

Special Access Rules

- Your organization must be using Professional, Enterprise, Developer, or Unlimited Edition and be enabled for the translation workbench.
- To view this object, you must have the "View Setup and Configuration" permission.

Fields

Field	Field Type	Field Properties	Description
LanguageLocaleKey	picklist	Create	The locale, which controls the language for labels displayed in an application. String is 2-5 characters long. The first two characters are always an ISO language code, for example "fr" or "en." If the value is further qualified by country, then the string also has an underscore (_) and another ISO country code, for example "US" or "UK. For example, the string for the United States is "en_US", and the string for French Canadian is "fr_CA."
		Filter	
		Nillable	
		Restricted picklist	
			For a list of the languages that Salesforce supports, see the Salesforce online help topic "What languages does Salesforce support?"
ScontrolId	reference	Create	The ID of the Scontrol that is being translated.
		Filter	
		Nillable	
Value	string	Create	The actual translated field label of the s-control. Label is Translation .
		Filter	
		Nillable	
		Update	

Usage

Use this object to translate your s-controls into the different languages supported by Salesforce. Users with the translation workbench enabled can view s-control translations, but either the "Customize Application" or "Manage Translation" permission is required to Create() or Update() s-control translations.
SelfServiceUser

Represents a Self-Service user. For details on how to implement Self-Service users, download the *Self-Service Implementation Guide* at http://nal.salesforce.com/help/doc/en/salesforce_selfservice_implementation_guide.pdf. Also, see "Setting Up Self-Service" in the Salesforce online help.

Corresponds to an SObject4 in which the ObjectType="SelfServiceUser".

Supported Calls

GetUpdated(),Login(),GetServerTimestamp()

Special Access Rules

Customer Portal users cannot access this object.

Field	Field Type	Field Properties	Description
ContactId	reference	Create Filter	Required. All Self-Service users must be associated with a Contact. The contact's email should match the Self-Service user email. The contact must have a value in the AccountID field or an error occurs. For information on IDs, see ID Field Type.
Email	email	Create Filter Update	Required. Make this the same as the email address for the Contact associated with this SelfServiceUser. Password resets and other system communication will be sent to this email address.
FirstName	string	Create Filter Nillable Update	First name of the Self-Service user.
IsActive	boolean	Create Defaulted on create Filter Update	Indicates whether the Self-Service user is allowed to log in to the Self-Service portal (true) or not (false). Note that there is no way to delete a Self-Service user. They can only be marked as inactive.
IsDeleted	boolean	Defaulted on create Filter	Indicates whether the record has been moved to the Recycle Bin (true) or not (false). Label is Deleted .

Field	Field Type	Field Properties	Description
LanguageLocaleKey	picklist	Create	Required. This is a restricted picklist field. It is the primary
		Filter	language for the user. All on-screen text in the Self-Service portal is displayed in this language.
		Restricted picklist	Fortan in and hay on the time tangangoi
		Update	
LastLoginDate	dateTime	Filter	The date and time when the Self-Service user last logged in.
		Nillable	
LastName	string	Create	Required. Last name of the Self-Service user.
		Filter	
		Update	
LocaleSidKey	picklist	Create	Required. This is a restricted picklist field. The value of this
		Filter	field affects the formatting and parsing of values, especially numeric values, in the Self-Service portal. Values are
		Restricted picklist	two-letter codes that indicate language and sometimes language and country. The codes are based on ISO standards.
		Update	
Name	string	Filter	Concatenation of FirstName and LastName. Limited to 121 characters.
SuperUser	boolean	Defaulted	Indicates whether this Self-Service user is a super user with
		Filter	additional access on his or her company's Self-Service portal
		Create	
TimeZoneSidKey	picklist	Create	a affects the offset used when displaying or entering times in
		Filter	the Self-Service portal.
		picklist	
		Update	
Username	string	Create	Required. This contains the name that a enters to log into
	Update	Filter	the Self-Service portal. Value must be unique in your organization. If you try to create or undate a with a duplicate
		Update	value, the operation is rejected and an error is returned.

For security reasons, you cannot query Self-Service user passwords via the API or the Salesforce user interface. However, the API allows you to set and reset Self-Service user passwords using the SetPassword() and ResetPassword() calls.

SelfServiceUser objects created from the API do not cause a notification email to be sent. If you want to notify the user, you must send them an email after creating the user.

Solution

Represents a detailed description of a customer issue and the resolution of that issue.

Corresponds to an SObject4 in which the ObjectType="Solution".

Supported Calls

Create(),Update(),DescribeSObjects(),Query(),Search(),Retrieve(),CreateObject(), GetDeleted(),GetUpdated()

Field	Field Type	Field Properties	Description
IsDeleted	boolean	Defaulted on create	Indicates whether the Solution has been moved to the Recycle Bin (true) or not (false). Label is Deleted .
		Filter	
IsHtml	boolean	Defaulted on create	Indicates whether the Solution is an HTML solution (true) or not (false).
		Filter	
IsOutOfDate	boolean	Defaulted on create	Read-only field that indicates whether a solution master has been updated since the translated version was created (true)
		Filter	or not (false). Note that this field does not appear in the page layout of master solutions.
IsPublished	boolean	Create	Indicates whether the Solution has been published (true)
		Defaulted on create	or not (false). A solution's published state does not affect how it can be used, or whether you can Query(),
		Filter	see the Salesforce online help. Label is Visible in Self-Service
		Update	Portal.
IsPublishedInPublicKb	boolean	Create	Indicates whether the Solution has been published in the
		Defaulted on create	Visible in Public Knowledge Base (true) or not (talse). Ladei 18
		Filter	
		Update	
IsReviewed	boolean	Defaulted on create	Indicates whether the Solution has been reviewed (true) or not (false). This flag can only be set indirectly via the
		Filter	Status picklist. Each predefined Status value implies an IsReviewed value. Label is Reviewed .
OwnerId	reference	Create	ID of the User who owns the Solution. For information on
		Defaulted on create	IDs, see ID Field Type.

Field	Field Type	Field Properties	Description
		Filter	
		Update	
ParentId	reference	Create	ID of the master solution, if this is the translation of a master
		Defaulted on create	solution.
		Filter	
		Update	
RecordTypeId	reference	Create	ID of the RecordType to which the Solution is associated.
		Defaulted on create	
		Filter	
		Update	
SolutionLanguage	picklist	Create	The language that the solution is written in, such as "French"
		Filter	or "Chinese (Traditional)."
		Restricted picklist	
		Update	
SolutionName	string	Create	Required. If a client application creates a new Solution
		Filter	a value for this field is unspecified, a hyphen (-), the default value for this field, is used. Limit: 255 characters. Label is
		Update	Title.
SolutionNote	textarea	Create	The details of the Solution record. Limit: 32,000 characters.
		Nillable	Label is Solution Details.
		Update	Note: If you have HTML Solutions enabled, any HTML tags used in this field are verified before the object is created or updated. If invalid HTML is entered, an error is thrown. Any JavaScript used in this field is removed before the object is created or updated. For more information, see "What are HTML Solutions?" in the Salesforce online help.
SolutionNumber	string	Autonumber	An identifying number that is assigned automatically when a solution is created. It cannot be set directly and it connect
	De	Defaulted on create	be modified.
		Filter	
Status	picklist	Create	Required. The status of the solution. Directly controls the
	F U	Filter	Is Reviewed flag. Io obtain the status values in the picklist, a client application can invoke the Query() call on the
		Update	SolutionStatus object.

Field	Field Type	Field Properties	Description
TimesUsed	int	Filter	Number of times this solution has been used. Label is Num Related Case.

Use this object to manage your organization's solution knowledge base. Client applications can Create(), Update(), and Query() Attachments associated with a solution.

SolutionHistory

Represents the history of changes to the values in the fields of a solution.

Supported Calls

Query(),Retrieve(),GetDeleted(),GetUpdated(),and DescribeSObjects()

Fields

Field	Field Type	Field Properties	Description
Field	picklist	Filter Restricted picklist	The name of the field that was changed. Label is Custom Field Definition ID.
IsDeleted	boolean	Defaulted on create Filter	Indicates whether the record has been moved to the Recycle Bin (true) or not (false). Label is Deleted .
SolutionId	reference	Filter	ID of the Solution. For information on IDs, see ID Field Type. Label is Solution ID .

Usage

Use this read-only object to identify changes to a solution.

This object respects field-level security on the parent object.

SolutionStatus

Represents the status of a Solution, such as Draft, Reviewed, and so on.

Corresponds to an SObject4 in which the ObjectType="SolutionStatus".

Supported Calls

Query(),Retrieve(),CreateObject(),DescribeSObjects()

Fields

For a complete list of fields in this object type, see the Salesforce online help. A client application can also inspect an SObject4 of this type, iterating through the array of Field4 objects in the Fields property.

Field	Field Type	Field Properties	Description
IsDefault	boolean	Defaulted on create Filter	Indicates whether this is the default solution status value (true) or not (false) in the picklist. Only one value can be the default value.
IsReviewed	boolean	Defaulted on create Filter	Indicates whether this solution status value represents a reviewed Solution (true) or not (false). Multiple solution status values can represent a reviewed Solution.
MasterLabel	string	Filter Nillable	Master label for this solution status value. This display value is the internal label that does not get translated.
SortOrder	int	Filter Nillable	Number used to sort this value in the solution status picklist. These numbers are not guaranteed to be sequential, as some previous solution status values might have been deleted.

Usage

This object represents a value in the solution status picklist. The solution status picklist provides additional information about the status of a Solution, such as whether a given status value represents a reviewed or unreviewed solution. Your client application can invoke the Query() call on this object to retrieve the set of values in the solution status picklist, and then use that information while processing Solution objects to determine more information about a given solution. For example, the application could test whether a given case has been reviewed or not based on its Status value and the value of the IsReviewed property in the associated SolutionStatus object.

This object is read-only. With sufficient permissions, your client application can invoke the Query() and DescribeSObjects() calls on these objects. You cannot Create(), Update() these objects.

SolutionTag

Associates a word or short phrase with a Solution.

Supported Calls

Create(),Query(),Retrieve(),DescribeSObjects()

Fields

Field	Field Type	Field Properties	Description
ItemId	reference	Create Filter	ID of the tagged item.
Name	string	Create Filter	Name of the tag. If this value does not already exist, a new TagDefinition is created and becomes the parent of this Tag object. Otherwise, a TagDefinition with the same name becomes the parent of this Tag object. Parent relationships are created automatically.
TagDefinitionId	reference	Filter	ID of the parent TagDefinition object that owns the tag.
Туре	picklist	Create Filter Restricted picklist	 Defines the visibility of a tag. Possible value are: Public: The tag can be viewed and manipulated by all users in an organization Personal: The tag can be viewed or manipulated only by a user with a matching OwnerId

Usage

SolutionTag stores the relationship between its parent TagDefinition and the Solution being tagged. Tag objects act as metadata, allowing users to describe and organize their data.

When a tag is deleted, its parent TagDefinition will also be deleted if the name is not being used; otherwise, the parent remains. Deleting a TagDefinition sends it to the recycle bin, along with any associated tag entries.

For more information on tags, see "About Tagging" in the Salesforce online help.

StaticResource

Represents a static resource that can be used in Visualforce markup. See "What is a Static Resource?" in the Salesforce online help.

Corresponds to an SObject4 in which the ObjectType="StaticResource".

Supported Calls

```
Create(),Update(),DescribeSObjects(),Query(),Retrieve(),CreateObject(),GetDeleted(),
GetUpdated()
```

Fields

Field	Field Type	Field Properties	Description
Body	base64	Create	Required. Encoded file data.
		Update	conversion of Base64 data between binary and String formats.
BodyLength	int	Filter	Size of the file (in bytes).
ContentType	string	Create	Type of content. Label is Mime Type . Limit: 120 characters.
		Filter	
		Update	
Description	textarea	Create	Text description of the static resource. Limit: 255 characters.
		Filter	
		Nillable	
		Update	
Name	string	Create	Required. Name of the static resource.
		Filter	
		Update	
NamespacePrefix	string	Create	The namespace prefix assigned to this object when it was
		Filter	created as part of a managed package. Null if this object is not part of a managed package. Limit is 15 characters. For
Nillable	Nillable	more information about managed packages and namespace prefixes, see "About Managed Packages" in the Salesforce online help.	
			This field cannot be accessed unless the logged-in user has the "Customize Application" permission.

Usage

Use static resources to upload content that you can reference in Visualforce markup, including archives (such as .zip and .jar files), images, stylesheets, JavaScript, and other files. Using a static resource is preferable to uploading a file to the Documents tab because:

- You can package a collection of related files into a directory hierarchy and upload that hierarchy as a .zip or .jar archive.
- You can reference a static resource in page markup by name using the *SResource* global variable instead of hard-coding document IDs.

Encoded Data

The API sends and receives the binary file data encoded as a base64 data type. Prior to Create(), clients must encode the binary file data as base64. Upon receiving an API response, clients must decode the base64 data to binary.

Maximum Static Resource Size

The Create () and Update () calls restrict static resources to a maximum size of 5 MB. An organization can have up to 250 MB of static resources, total.

TagDefinition

Defines the attributes of child Tag objects.

Supported Calls

DescribeSObjects(),Query(),Retrieve(),Search(),Undelete(),Update()

Fields

Field	Field Type	Field Properties	Description
Name	string	Filter	Identifies the tag word or phrase.
		Nillable	
		Update	
Туре	picklist	Filter	Defines the visibility of a tag. Possible
		Nillable	Public: The tag can be viewed and
		Restricted picklist	manipulated between all users in an organization
			• Personal: The tag can be viewed or manipulated only by a user with a matching OwnerId

Usage

When you create a tag for a record, an association is created with to a corresponding TagDefinition:

- If the value in the tag's Name field is new, a new TagDefinition record is automatically created and becomes the parent of the tag.
- If the value in the tag's Name field already exists in a TagDefinition, that TagDefinition automatically becomes the parent of the tag.

Each TagDefinition record has a one-to-many relationship with its child tag records.

The following standard objects represent tags for records:

- AccountTag
- AssetTag
- CampaignTag
- CaseTag
- ContactTag
- ContractTag
- DocumentTag

- EventTag
- LeadTag
- NoteTag
- OpportunityTag
- SolutionTag
- TaskTag

Custom objects may also be tagged. Tags for custom objects are identified by a suffix of two underscores immediately followed by the word tag. For example, a custom object named Meeting has a corresponding tag named Meeting_tag in that organization's WSDL. Meeting_tag is only valid for Meeting objects.

TagDefinition is useful for mass operations on any tag record. For instance, if you want to rename existing tags, you can search for the appropriate TagDefinition object, update it, and the child tag's Name values are also changed. The following Java example replaces all WC tags with the phrase West Coast:

```
public void tagDefinitionSample() {
    QueryResult qr = null;
    try {
        qr = binding.query("SELECT Id, Name FROM TagDefinition WHERE Name = 'WC'");
    } catch (RemoteException e) {
        System.out.println("An unexpected error has occurred." + e.getMessage());
    }
    TagDefinition d = (TagDefinition)qr.getRecords()[0];
    d.setName("West Coast");
    try {
        binding.update(new SObject[]{d});
    } catch (RemoteException e) {
        System.out.println("An unexpected error has occurred." + e.getMessage());
    }
}
```

When a tag is deleted, its parent TagDefinition will also be deleted if the name is not being used; otherwise, the parent remains. Deleting a TagDefinition sends it to the recycle bin, along with any associated tag entries.

For more information on tags, see "About Tagging" in the Salesforce online help.

Task

Represents a business activity such as making a phone call or other "to-do" items. In the Salesforce user interface, task and Event records are collectively referred to as activities.

Corresponds to an SObject4 in which the ObjectType="Task".



Note: Task fields related to calls are exclusive to Salesforce CRM Call Center.

Supported Calls

```
Create(),Update(),DescribeSObjects(),Query(),Search(),Retrieve(),CreateObject(),
GetDeleted(),GetUpdated()
```

Field	Field Type	Field Properties	Description
AccountId	reference	Filter Nillable	ID of the associated Account. For information on IDs, see ID Field Type.
ActivityDate	date	Create	The due date of the task. This field has a timestamp that is
		Filter	(UTC) time zone. The timestamp is not relevant; do not
		Nillable	attempt to alter it in order to accommodate time zone
		Update	differences. Laber is Due Date .
CallDisposition	string	Create	Represents the result of a given call, for example, "we'll call
		Filter	back," or "call unsuccessful." Limit is 255 characters.
		Nillable	Not subject to field-level security, available for any user in
		Update	an organization with Salesforce CRM Call Center.
CallDurationInSeconds	int	Create	Duration of the call in seconds.
		Filter	Not subject to field-level security, available for any user in
		Nillable	an organization with Salesforce CRM Call Center.
		Update	
CallObject	string	Create	Name of a call center. Limit is 255 characters.
		Filter	Not subject to field-level security, available for any user in
		Nillable	an organization with Salesforce CRM Call Center.
		Update	
CallType	picklist	Create	The type of call being answered: Inbound, Internal, or
		Filter	Outbound.
		Nillable	
		Update	
ConnectionReceivedID	reference	Filter	ID of the PartnerNetworkConnection that shared this record
		Nillable	with your organization. This field is only available if you have enabled Salesforce to Salesforce.
ConnectionSentID	reference	Filter	ID of the PartnerNetworkConnection that you shared this
		Nillable	Salesforce to Salesforce. Beginning with API version 15.0, the ConnectionSentID field is no longer supported. The ConnectionSentID field is still be visible, but the value is null. You can use the new PartnerNetworkRecordConnection object to forward records to connections.
Description	textarea	Create	Text description of the task.
		Nillable	

Task

Field	Field Type	Field Properties	Description
		Update	
IsClosed	boolean	Defaulted on create	Indicates whether the task was completed (true) or not (false). Is only set indirectly via the Status picklist. Label
		Filter	is Closed .
IsDeleted	boolean	Defaulted on create	Indicates whether the record has been moved to the Recycle Bin (true) or not (false).
		Filter	Label is Deleted .
IsReminderSet	boolean	Create	Indicates whether a popup reminder has been set for the task
		Defaulted on create	(true) or not (false).
		Filter	
		Update	
IsVisibleInSelfService	boolean	Defaulted on create	Indicates whether the task can be viewed in the Customer Self-Service Portal (true) or not (false).
		Filter	
OwnerId	reference	Create	ID of the User who owns the record. For information on
		Defaulted on create	IDs, see ID Field Type. Label 1s Assigned To ID.
		Filter	
		Update	
Priority	picklist	Create	Required. Indicates the importance or urgency of a task, such
		Filter	as high or low.
		Update	
ReminderDateTime	dateTime	Create	Represents the time the reminder is scheduled to fire, if
		Filter	the user may have deselected the reminder checkbox in the
		Nillable	Salesforce user interface, or the reminder has already fired at
		Update	the time indicated by the value.
Status	picklist	Create	Required. The current status of the task, such as In Progress
		Filter	for the Isclosed flag. To obtain picklist values, a client
		Update	application can invoke the Query () call on the TaskStatus object.
Subject	comobobox	Create	The subject line of the task, such as "Call" or "Send Quote."
		Filter	
		Nillable	

Field	Field Type	Field Properties	Description
		Update	
WhatId	reference	Create Filter Nillable Update	ID of a related Account, Opportunity, Campaign, Case, or custom object. Label is Opportunity/Account ID .
WhoId	reference	Create Filter Nillable Update	ID of a related Contact or Lead. If the Whold refers to a lead, then the Whatld field must be empty. Label is Contact/Lead ID.

For information about working with archived tasks, see Archived Activities.

TaskPriority

Represents the importance or urgency of a Task, such as High, Normal, or Low.

Corresponds to an SObject4 in which the ObjectType="TaskPriority".

Supported Calls

Query(),Retrieve(),CreateObject(),DescribeSObjects()

Special Access Rules

Customer Portal users cannot access this object.

Fields

For a complete list of fields in this object type, see the Salesforce online help. A client application can also inspect an SObject4 of this type, iterating through the array of Field4 objects in the Fields property.

Field	Field Type	Field Properties	Description
IsDefault	boolean	Defaulted on create Filter	Indicates whether this is the default task priority value (true) or not (false) in the picklist. Only one value in the picklist can be the default value.
IsHighPriority	boolean	Defaulted on create Filter	Indicates whether this task priority value represents a high priority Task(true) or not(false). Multiple task priority values can represent a high-priority Task.

Field	Field Type	Field Properties	Description
MasterLabel	string	Filter Nillable	Master label for this task priority value. This display value is the internal label that does not get translated. Limit: 255 characters.
SortOrder	int	Filter Nillable	Number used to sort this value in the task priority picklist. These numbers are not guaranteed to be sequential, as some previous task priority values might have been deleted.

This object represents a value in the task priority picklist. The task priority picklist provides additional information about the importance of a Task, such as whether a given priority value represents a high priority. Your client application can invoke the Query () call on this object to retrieve the set of values in the task priority picklist, and then use that information while processing Task objects to determine more information about a given task. For example, the application could test whether a given Task is high priority based on its Priority value and the value of the IsHighPriority property in the associated TaskPriority object.

This object is read-only. With sufficient permissions, you can invoke the Query () and DescribeSObjects () calls on these objects.

TaskStatus

Represents the status of a Task, such as Not started, Completed, or Closed.

Corresponds to an SObject4 in which the ObjectType="TaskStatus".

Supported Calls

Query(),Retrieve(),CreateObject(),DescribeSObjects()

Special Access Rules

Customer Portal users cannot access this object.

Field	Field Type	Field Properties	Description
IsClosed	boolean	Defaulted on create Filter	Indicates whether this task status value represents a closed Task (true) or not (false). Multiple task status values can represent a closed Task.
IsDefault	boolean	Defaulted on create Filter	Indicates whether this is the default task status value (true) or not (false) in the picklist.

Field	Field Type	Field Properties	Description
MasterLabel	string	Filter Nillable	Master label for this task status value. This display value is the internal label that does not get translated. Limit: 255 characters.
SortOrder	int	Filter Nillable	Number used to sort this value in the task status picklist. These numbers are not guaranteed to be sequential, as some previous task status values might have been deleted.

This object represents a value in the task status picklist. The task status picklist provides additional information about the status of a Task, such as whether a given status value represents an open or closed task. Your client application can invoke the Query() call on this object to retrieve the set of values in the task status picklist, and then use that information while processing Task objects to determine more information about a given task. For example, the application could test whether a given task is open or closed based on the Task Status value and the value of the Isclosed property in the associated TaskStatus object.

This object is read-only. With sufficient permissions, your client application can invoke the Query() and DescribeSObjects() calls on these objects.

TaskTag

Associates a word or short phrase with a Task.

Supported Calls

Create(),Query(),Retrieve(),DescribeSObjects()

Field	Field Type	Field Properties	Description
ItemId	reference	Create Filter	ID of the tagged item.
Name	string	Create Filter	Name of the tag. If this value does not already exist, a new TagDefinition is created and becomes the parent of this Tag object. Otherwise, a TagDefinition with the same name becomes the parent of this Tag object. Parent relationships are created automatically.
TagDefinitionId	reference	Filter	ID of the parent TagDefinition object that owns the tag.
Туре	picklist	Create Filter Restricted picklist	 Defines the visibility of a tag. Possible value are: Public: The tag can be viewed and manipulated by all users in an organization Personal: The tag can be viewed or manipulated only by a user with a matching OwnerId

TaskTag stores the relationship between its parent TagDefinition and the Task being tagged. Tag objects act as metadata, allowing users to describe and organize their data.

When a tag is deleted, its parent TagDefinition will also be deleted if the name is not being used; otherwise, the parent remains. Deleting a TagDefinition sends it to the recycle bin, along with any associated tag entries.

For more information on tags, see "About Tagging" in the Salesforce online help.

Territory

Represents a flexible collection of accounts and users where the users have at least read access to the accounts, regardless of who owns the accounts. Only available if territory management has been enabled for your organization. For more information, see the "What is Territory Management?" topic in the Salesforce online help.

Supported Calls

Create(),Query(),Retrieve(),GetDeleted(),GetUpdated(),GetUpdated(),DescribeSObjects()

Fields

I

Field	Field Type	Field Properties	Description
AccountAccessLevel	picklist	Create	Account access level granted to users assigned to this territory.
		Filter	
		Restricted picklist	
		Update	
CaseAccessLevel	picklist	Create	Case access level granted to users assigned to this territory.
		Filter	
		Nillable	
		Restricted picklist	
		Update	
ContactAccessLevel	picklist	Create	A value that represents the type of access granted to the target
		Filter	Group, UserRole, or User for any associated contacts. The possible values are:
		Restricted	• None
		picklist	• Read
		Update	• Edit
			Note: When DefaultContactAccess is set to "Controlled by Parent," you cannot create or update this field.

Field	Field Type	Field Properties	Description
Description	string	Create	A description of the territory that is 1,000 characters or less.
		Filter	
		Nillable	
		Update	
ForecastUserId	reference	Create	ID of the Forecast Manager, who is the user to whom
		Filter	information on IDs, see ID Field Type. Label is User ID.
		Nillable	
		Update	
MayForecastManagerShare	boolean	Filter	Indicates whether the forecast manager can manually share their own forecast.
Name	string	Create	A name for the territory. Limit is 80 characters. Label is
		Filter	Territory Name.
		Update	
OpportunityAccessLevel	picklist	Create	Opportunity access level granted to users assigned to this
		Filter	territory.
		Restricted picklist	
		Update	
ParentTerritoryID	reference	Create	Territory immediately above this territory in the territory
		Filter	hierarchy. Label is Parent Territory ID .
		Nillable	
		Update	
RestrictOppTransfer	boolean	Create	Indicates whether the opportunities associated with this
		Defaulted on create	territory are kept within the bounds of this territory and this territory's children when account assignment rules are run (true), or if opportunities associated with this territory can
		Filter	be assigned to other nodes of the territory hierarchy when
		Update	account assignment rules are run (false). Label is Confine Opportunity Assignment .

Use the Territory object to Query () your organization's territory hierarchy. Use it to obtain valid territory IDs when querying or modifying objects associated with territories.

User

Represents a user in your organization.

Corresponds to an SObject4 in which the ObjectType="User".

Supported Calls

```
Create(),Update(),Query(),Search(),Retrieve(),CreateObject(),GetUpdated(),
DescribeSObjects()
```

Special Access Rules

- To Create () or Update () a User object, you must log in with the "Manage Users" permission.
- Customer Portal users can only view Custom Portal users in that account.

Field	Field Type	Field Properties	Description
Alias	string	Create	Required. The user's alias. For example, "jsmith."
		Filter	
		Update	
CallCenterId	reference	Create	If Salesforce CRM Call Center is enabled,
		Filter	represents the call center to which this user is assigned.
		Nillable	0
		Update	
City	string	Create	The city associated with the user.
		Filter	
		Nillable	
		Update	
CommunityNickname	string	Create	Name used to identify this user in the Salesforce
		Filter	CRM Ideas online community.
		Nillable	
		Update	
CompanyName	string	Create	The name of the user's company.
		Filter	
		Nillable	
		Update	

Field	Field Type	Field Properties	Description
ContactId	reference	Create Filter	ID of the Contact associated with this account. The contact must have a value in the AccountID field or an error occurs. For information on IDs, see ID Field Type.
Country	string	Create	The country associated with the user.
		Filter	
		Nillable	
		Update	
DefaultCurrencyIsoCode	picklist	Create	The user's default currency setting for new records.
		Filter	For example, a user in France could have a
		Nillable	that would be their default currency in the
		Restricted picklist	application. However, the User object could have currency custom fields stored in a different currency.
		Update	Only applicable for organizations that use multiple currencies.
			For more information, see CurrencyIsoCode
DefaultDivision	picklist	Create	The user's default division. Only applicable if
		Defaulted on create	divisions are enabled for your organization. For more information, see the Salesforce online help.
		Filter	
		Restricted	
		Update	
DelegatedApproverId	reference	Create	Id of the user who is a delegated approver for this
		Filter	user.
		Nillable	
		Update	
Department	string	Create	The company department associated with the user.
		Filter	
		Nillable	
		Update	
Division	string	Create	The division associated with this user, similar to
		Filter	Department and unrelated to
			DefaultDivision.
		Nillable	

Field	Field Type	Field Properties	Description
Email	email	Create Filter	Required. The user's email address.
		Update	
EmailEncodingKey	picklist	Create	Required. The email encoding for the user, such
		Filter	as 150-8859-1 or 01F-8.
		Restricted picklist	
		Update	
EmployeeNumber	string	Create	The user's employee number.
		Filter	
		Nillable	
		Update	
Extension	phone	Create	The user's phone extension number.
		Filter	
		Nillable	
		Update	
Fax	phone	Create	The user's fax number.
	1	Filter	
		Nillable	
		Update	
FederationIdentifier	string	Create	Indicates the value that must be listed in the
		Filter	Subject element of a Security Assertion Markup Language (SAML) <i>IDP certificate</i> to authenticate the user for a client application using single sign-on with Salesforce. This value must be specified if the SAML USET ID Type is Assertion contains Federation ID from the User object. Otherwise, this field cannot be edited.
FirstName	string	Create	The user's first name.
		Filter	
		Nillable	
		Update	
ForecastEnabled	boolean	Create	Indicates whether the user is enabled as a Forecast
		Defaulted on create	Manager (true) or not (false) in customizable forecasting. Forecast managers see forecast rollups from users below them in the forecast hierarchy
		Filter	about bolow chem in the forecast inerarchy.

Field	Field Type	Field Properties	Description
		Update	
IsActive	boolean	Create Defaulted on create Filter Update	Indicates whether the user has access to log in (true) or not (false). You can modify a User's active status from the Salesforce user interface or via the API.
IsPartner	boolean	Defaulted on create Filter	Indicates whether the user is a partner who has access to the PRM Portal (true) or not (false). This field is not available for release 9.0 and later. Instead, use UserType with the value PRM.
IsPortalSelfRegistered	boolean	Defaulted on create Filter	Indicates whether the user is a Customer Portal user who self-registered for your organization's Customer Portal (true) or not (false). This field is not available for release 9.0 and earlier. For more information on the Customer Portal and self-registration, see Enabling Customer Portal Login and Settings in the Salesforce online help.
LanguageLocaleKey	picklist	Create Filter Restricted picklist Update	Required. The user's language, such as "French" or "Chinese (Traditional)." Label is Language .
LastLoginDate	dateTime	Filter Nillable	The date and time when the user last logged in.
LastName	string	Create Filter Update	Required. The user's last name.
LocaleSidKey	picklist	Create Filter Restricted picklist Update	Required. This field is a restricted picklist field. The value of the field affects formatting and parsing of values, especially numeric values, in the user interface. It does not affect the API. The field values are named according to the language, and country if necessary, using two-letter ISO codes. The set of names is based on the ISO standard. It can often be more convenient to manually set a user's locale in the Salesforce user interface, and then use that value for inserting or updating other users via the API.

Field	Field Type	Field Properties	Description
Manager	picklist	Create	User lookup field used to select the user's manager. This establishes a hierarchical relationship
		Filter	preventing you from selecting a user that directly
		Restricted picklist	or indirectly reports to itself.
		Update	
MobilePhone	phone	Create	The user's mobile or cellular phone number.
		Filter	
		Nillable	
		Update	
MobileUser	boolean	Create	Indicates whether the user is allocated one
		Update	Force.com Mobile license (true) or not (false). Label is Mobile User . The Force.com Mobile license grants the user access to Force.com Mobile capabilities.
Name	string	Filter	Concatenation of FirstName and LastName. Limited to 121 characters.
OfflineTrialExpirationDate	dateTime	Filter	The date and time when the user's Connect Offline
		Nillable	trial expires.
Phone	phone	Create	The user's phone number.
		Filter	
		Nillable	
		Update	
PortalRole	picklist	Create	The role of the user in the Customer Portal (either
		Filter	Executive, Manager, User, or PersonAcount).
		Nillable	
PostalCode	string	Create	The user's postal or ZIP code. Label is Zip/Postal
		Filter	Code.
		Nillable	
		Update	
ProfileId	reference	Create	Required. ID of the user's Profile. Use this value
		Filter	to cache metadata based on profile. For information on IDs, see ID Field Type. In earlier
		Update	releases, this was RoleId.
ReceivesAdminInfoEmails	boolean	Create	Indicates whether the user receives email for
		Defaulted on create	administrators from salesforce.com (true) or not (false).

Field	Field Type	Field Properties	Description
		Filter	
		Update	
SFContentUser	boolean	Create	Indicates whether the user is allocated one
		Update	Salesforce CRM Content User License (true) or not (false). Label is Salesforce CRM User . The Salesforce CRM Content User license grants the user access to the Salesforce CRM Content application.
ReceiveInfoEmails	boolean	Create	Indicates whether the user receives informational
		Defaulted on create	email from salesforce.com (true) or not (false).
		Filter	
		Update	
State	string	Create	The state associated with the User.
		Filter	
		Nillable	
		Update	
Street	textarea	Create	The street address associated with the User.
		Filter	
		Nillable	
		Update	
TimeZoneSidKey	picklist	Create	Required. This field is a restricted picklist field. A
		Filter	User's time zone affects the offset used when displaying or entering times in the user interface.
		Restricted picklist	However, the API does not use a User's time zone when querying or setting values.
		Update	Values for this field are named using region and key city, according to ISO standards. It can often be more convenient to manually set a User's time zone in the user interface, and then use that value for creating or updating other Users via the API.
Title	string	Create	The user's business title, such as "Vice President."
		Filter	
		Nillable	
		Update	
Username	string	Create	Required. Contains the name that a user enters to
		Filter	log into the API or the Salesforce user interface. The value for this field must be in the form of an

Field	Field Type	Field Properties	Description
		Update	email address. It must also be unique across all Salesforce instances. If you try to Create() or Update() a User with a duplicate value for this field, the operation is rejected.
			Each inserted User also counts as a license in Salesforce. Every organization has a maximum number of licenses. If you attempt to exceed the maximum number of licenses by inserting User records, the Create() call is rejected.
UserPermissionsCallCenterAutoLogin	boolean	Create Update	Required if Salesforce CRM Call Center is enabled. Indicates whether the user is enabled to use the auto login feature of the call center (true) or not (false).
UserPermissionsMarketingUser	boolean	Create Update	Required. Indicates whether the user is enabled to manage campaigns in the online application (true) or not (false). Label is Marketing User .
UserPermissionsOfflineUser	boolean	Create Update	Required. Indicates whether the user is enabled to use Offline Edition (true) or not (false). Label is Offline User .
UserPreferencesActivityRemindersPopup	boolean	Filter Nillable	When true, a reminder popup window automatically opens when an activity reminder is due. Corresponds to the Trigger alert when reminder comes due checkbox at Setup > My Personal Information > Reminders in the Salesforce user interface. See "Customizing Activity Reminders" in the Salesforce online help.
UserPreferencesApexPagesDeveloperMode	boolean	Filter Nillable	When true, indicates the user has enabled developer mode for editing Visualforce pages and controllers.
UserPreferencesEventRemindersCheckboxDefault	boolean	Filter Nillable	When true, a reminder popup is automatically set on the user's events. Corresponds to the By default, set reminder on Events to checkbox Setup > My Personal Information > Reminders in the online application. This field is related to UserPreference. Also, see Customizing Activity Reminders in the Salesforce online help.
UserPreferencesTaskRemindersCheckboxDefault	boolean	Filter Nillable	When true, a reminder popup is automatically set on the user's tasks. Corresponds to the By default, set reminder on Tasks to checkbox at Setup ➤ My Personal Information ➤ Reminders in the online application. This field is related to UserPreference. Also, see Customizing Activity Reminders in the Salesforce online help.

Field	Field Type	Field Properties	Description
UserPreferencesReminderSoundOff	boolean	Filter Nillable	When true, a sound automatically plays when an activity reminder is due. Corresponds to the Play a reminder sound checkbox at Setup ➤ My Personal Information ➤ Reminders in the online application. See Customizing Activity Reminders in the Salesforce online help.
UserRoleId	reference	Create Filter Nillable Update	ID of the user's UserRole. Label is Role ID .
UserType	picklist	Filter Nillable Restricted picklist	 The category of user license. Each UserType is associated with one or more UserLicense records. Each UserLicense is associated with one or more profiles. In API version 10.0 and later, valid values include: Standard: Salesforce user license. This user type also includes Salesforce Platform and Salesforce Platform One user licenses. Label is Standard. PowerPartner: PRM user whose access is limited because he or she is a partner and typically accesses the application through a partner portal. Label is Partner. CustomerSuccess: user whose access is limited because he or she is an organization's customer and accesses the application through a customer portal. Label is Customer Portal User. PowerCustomerSuccess: user whose access is limited because he or she is an organization's customer and accesses the application through a customer portal. Label is Customer Portal User. Users with this license type can view and edit data they directly own or data owned by or shared with users below them in the customer portal role hierarchy.
WirelessEmail	email	Create Filter Nillable Update	Wireless email address associated with this user.

Use this object to query information about users and to provision and modify users in your organization. Unlike other objects, the records in the User table represent actual users—not data owned by users. All Users have access to use Query() or DescribeSObjects() with User objects.

Deactivate Users

You cannot delete Users in the Salesforce user interface or the API. You can deactivate a User in the Salesforce user interface. Because User can never be deleted, we recommend that you exercise caution when creating them.

Passwords

For security reasons, you cannot query User passwords via the API or the Salesforce user interface. However, the API allows you to set and "reset" User passwords using the SetPassword() and ResetPassword() calls. The password lockout status and the ability to reset the User locked-out status is not available via the API. You must check and reset the User password lockout status using the Salesforce user interface.

UserAccountTeamMember

Represents a single User on the default account team of another User.

Corresponds to an SObject4 in which the ObjectType="UserAccountTeamMember".

See also OpportunityTeamMember, which represents a User on the sales team of an Opportunity

Supported Calls

```
Create(),Update(),Query(),Search(),Retrieve(),CreateObject(),GetDeleted(),GetUpdated(),
DescribeSObjects()
```

Special Access Rules

Customer Portal users cannot access this object.

Field	Field Type	Field Properties	Description
AccountAccessLevel	picklist	Create Filter Restricted picklist Update	 Required. Level of access that the account team member has to Accounts that the user has added to his or her default account team. The possible values are: Read Edit This field must be set to an access level that is higher than the organization's default access level for accounts.
CaseAccessLevel	picklist	Create Filter	Required. Level of access that the account team member has to Cases related to the account. The possible values are: • None

Field	Field Type	Field Properties	Description
		Restricted picklist	• Read • Edit
		Update	This field must be set to an access level that is higher than the organization's default access level for cases.
ContactAccessLevel	picklist	Create Filter Restricted picklist Update	Required. Level of access that the account team member has to Contacts related to the account. The possible values are: • None • Read • Edit This field must be set to an access level that is higher than the organization's default access level for contacts. Note: When DefaultContactAccess is set to "Controlled by Parent," you cannot create or update this field.
OpportunityAccessLevel	picklist	Create Filter Restricted picklist Update	Required. Level of access that the team member has to Opportunity records related to the account. The possible values are: • None • Read • Edit This field must be set to an access level that is higher than the organization's default access level for opportunities.
OwnerId	reference	Create Filter	Required. ID of the User who owns the default account team. This field cannot be updated. For information on IDs, see ID Field Type.
TeamMemberRole	picklist	Create Filter Nillable Update	Role that the team member has on opportunities for which the user has added his or her default account team. The valid values are set by the organization's administrator in the Account Team Roles picklist. Label is Team Role .
UserId	reference	Create Filter	Required. ID of the User who is a member of the default account team. This field cannot be updated.

This object is available only in organizations that have enabled the account teams functionality, which can be done using the Salesforce user interface.

If you attempt to create a record that matches an existing record, the Create() call updates any modified fields and returns the existing record.

Users can set up their default account team to include the other Users that typically work with them on accounts.

See UserTeamMember for information about the object related to default sales teams.

UserLicense

Represents a user license in your organization. A user license entitles a user to different functionality within Salesforce and determines the profiles available to the user.

Supported Calls

Query(),Retrieve(),DescribeSObjects()

Field	Field Type	Field Properties	Description
Name	string	Filter	The internal name of the user license. Supported values are:
		Nillable	Salesforce Platform User License: label is Salesforce Platform
		Salesforce Platform One User License (1) : label is Salesforce Platform One	
			Full CRM: label is Salesforce
			Partner: label is Partner
			Basic Partner: label is Basic Partner
			Standard Partner: label is Standard Partner
			Strategic Partner: label is Strategic Partner
			Customer Portal Manager: label is Customer Portal Manager
			Customer Portal Manager Basic : label is Basic Customer Portal Manager
			Customer Portal Manager Standard : label is Standard Customer Portal Manager
			Customer Portal User: label is Customer Portal User
			Please note that your organization may also include custom user licenses.
LicenseDefinitionKey	string	Filter Nillable	A string that uniquely identifies a particular user license. Label is License Def. ID. Values are:
			AUL: corresponds to the Salesforce Platform user license
			AUL1: corresponds to the Salesforce Platform One user license
			SFDC: corresponds to the Salesforce user license
			POWER_PRM : corresponds to the Partner user license
			PID_BASIC_PRM : corresponds to the Basic Partner user license

Field	Field Type	Field Properties	Description
			PID_STANDARD_PRM : corresponds to the Standard Partner user license
			PID_STRATEGIC_PRM : corresponds to the Strategic Partner user license
			POWER_SSP : corresponds to the Customer Portal Manager user license
			PID_Customer_Portal_Basic: corresponds to the Customer Portal Manager Standard user license
			PID_Customer_Portal_Standard : corresponds to the Standard Customer Customer Portal user license
			SSP: corresponds to the Customer Portal User user license
			PID_Limited_Customer_Portal_Basic : corresponds to the Customer Portal Manager Standard user license
			PID_Overage_Customer_Portal_Basic: corresponds to the Customer Portal Manager Standard user license

Users with the "View Setup and Configuration" permission can use the UserLicense object to view the set of currently defined user licenses in your organization.

The UserLicense object is currently used by bulk user creation to determine the user license to which each profile belongs. For example, if you use the API to create portal users and you want to know which profile belongs to each portal user license, you can Query() this object for each profile and check the LicenseDefinitionKey to identify the associated user license.

UserPreference

Represents a functional preference for a specific user in your organization.

Corresponds to an SObject4 in which the ObjectType="UserPreference".

Supported Calls

```
Query(),Retrieve(),DescribeSObjects()
```

Special Access Rules

Customer Portal users cannot access this object.

Fields

Field	Field Type	Field Properties	Description
Preference	picklist	Filter	The name of the user preference. Supported values are Event Reminder Default Lead Time and Task Reminder Default Time. These values are related to UserPreferencesEventRemindersCheckboxDefault and UserPreferencesTaskRemindersCheckboxDefault on the User object.
UserId	reference	Filter	The ID of the user associated with this role. Label is User ID.
Value	string	Filter Nillable	The value of the user preference. For Event Reminder Default Lead Time, the values are increasing intervals of time from 0 minutes to 2 days. For Task Reminder Default Time, the values are half-hours from 12:00 AM to 11:30 PM. To view the respective sets of values, go to Setup ➤ My Personal Information ➤ Reminders in the online application.

Usage

Use this object to Query () the set of currently configured user preferences in your organization. In your client application, you can query the User object to obtain valid User IDs in order to access the UserPreference object.

All Users have access to invoke Query () or DescribeSObjects () with this object.

UserRole

Represents a user role in your organization.

Corresponds to an SObject4 in which the ObjectType="UserRole".

Note: This object was called "Role" in previous versions of the API documentation.

Supported Calls

```
Create(),Update(),DescribeSObjects(),Query(),Search(),Retrieve(),CreateObject(),
GetDeleted(),GetUpdated()
```

Special Access Rules

Customer Portal users cannot access this object.

Field	Field Type	Field Properties	Description
CaseAccessForAccountOwner	picklist	Filter	The case access level for the account owner.
		Nillable	
		Restricted picklist	
ContactAccessForAccountOwner	picklist	Filter	The contact access level for the account owner.
		Nillable	Note: When DefaultContactAccess is
		Restricted picklist	set to "Controlled by Parent," you cannot create or update this field.
ForecastUserId	reference	Filter	The ID of the forecast manager associated with this
		Nillable	role. Label is User ID .
IsPartner	boolean	Defaulted on create	Indicates whether the user role is a partner who has access to the PRM Portal (true) or not (false).
		Filter	This field is not available for release 9.0 and later. Instead, use PortalType with the value Partner.
MayForecastManagerShare	boolean	Filter	Indicates whether the forecast manager can manually share their own forecast.
Name	string	Filter	Required. Name of the role.
OpportunityAccessForAccountOwner	picklist	Filter	Required. The opportunity access level for the
	Restricted picklist account owner. N picklist organization-wid	account owner. Note that you cannot set a user role with an opportunity access less than that specified in organization-wide defaults.	
ParentRoleId	reference	Filter	The ID of the parent role.
		Nillable	
PortalRole	picklist	Filter	The portal role (either Executive, Manager, User, or
		Nillable	PersonAccount).
PortalType	picklist	Defaulted	This value indicates the type of portal for the role:
		on create	None: Salesforce application role.
		Filter	 Partner: PRM Portal role. The field IsPartner used in release 8.0 will map to this value.
			This field replaces IsPartner beginning with release 9.0.
RollupDescription	string	Filter	Description of the forecast rollup. Label is
		Nillable	Description.

Use this object to Query () the set of currently configured user roles in your organization. Use it in your client application to obtain valid UserRole IDs to use when querying or modifying a User.

All users have access to invoke Query() or DescribeSObjects() with this object. If your client application logs in with the "Modify All Data" permission, it can Query(), Create(), and Update() UserRole records.

UserTeamMember

Represents a single User on the default sales team of another User.

Corresponds to an SObject4 in which the ObjectType="UserTeamMember".

Supported Calls

```
Create(),Update(),Query(),Search(),Retrieve(),CreateObject(),GetDeleted(),GetUpdated(),
DescribeSObjects()
```

Special Access Rules

- This object is available only in organizations that have enabled the team selling functionality.
- Customer Portal users cannot access this object.

Field	Field Type	Field Properties	Description
OpportunityAccessLevel	tunityAccessLevel picklist Create Filter Restricted		Required. Level of access that the team member has to opportunities for which the user has added his or her default sales team. The possible values are: • Read
		picklist Update	• Edit This field must be set to an access level that is higher than the organization's default access level for opportunities.
OwnerId	reference	Create Filter	Required. ID of the User who owns the default sales team. This field cannot be updated. For information on IDs, see ID Field Type.
TeamMemberRole	picklist	Create Filter Nillable Update	Role that the team member has on opportunities for which the User has added his or her default sales team. The valid values are set by the organization's administrator in the Sales Team Roles picklist. Label is Team Role .
UserId	reference	Create Filter	Required. ID of the User who is a member of the default sales team. This field cannot be updated.

If you attempt to create a record that matches an existing record, the Create() call updates any modified fields and returns the existing record.

User objects can set up their default sales team to include the other User objects that typically work with them on opportunities.

UserTerritory

Represents a User who has been assigned to a Territory. For more information, see the "What is Territory Management?" topic in the Salesforce online help.

Supported Calls

Create(),Query(),Retrieve(),GetDeleted(),GetUpdated(),DescribeSObjects()

Special Access Rules

- Only available if territory management has been enabled for your organization.
- · Customer Portal users cannot access this object.

Field	Field Type	Field Properties	Description
IsActive	boolean	Defaulted on create Filter	 Indicates whether the user is active in the given territory (true), or inactive in the given territory (false): Users who are active in a territory are explicitly assigned to the territory and can have open opportunities, closed opportunities, or no opportunities associated with that territory. Users who are inactive in a territory are not explicitly assigned to the territory, but own an open or closed opportunity that is associated with the territory. For example, a user may have been transferred out of a territory, but still own opportunities in his or her old territory.
			Until a user is deleted from a territory (not simply removed from the territory), the record will not be returned in a GetDeleted() call.
IsDeleted	boolean	Defaulted on create Filter	Indicates whether the record has been moved to the Recycle Bin (true) or not (false). Label is Deleted .
TerritoryId	reference	Create Filter	ID of the Territory to which the user has been assigned. For information on IDs, see ID Field Type.

Field	Field Type	Field Properties	Description
UserId	reference	Create	ID of the user.
		Filter	

If a user is inactive in a territory, and the opportunities they own that are associated with the territory are all closed, then the user is not returned.

Vote

Represents a vote that a user has made on an idea.

Corresponds to an SObject4 in which the ObjectType="Vote".

Supported Calls

Create(),Query(),Retrieve(),CreateObject(),DescribeSObjects(),GetDeleted(),GetUpdated()

Fields

Field	Field Type	Field Properties	Description
IsDeleted	boolean	Defaulted on create Filter	Indicates whether the object has been moved to the Recycle Bin (true) or not (false). Label is Deleted .
ParentId	reference	Create Filter	ID of the Idea associated with this vote. For information on IDs, see ID Field Type.
Туре	picklist	Create Filter Restricted picklist	Picklist that indicates the type of vote. The value Up indicates that the vote is a user's positive endorsement of the associated idea. The value Down indicates that the vote is a user's negative endorsement of the associated idea.



Note: If you are importing Vote data into Salesforce and need to set the value for an audit field, such as CreatedDate, contact salesforce.com. Audit fields are automatically updated during API operations unless you request to set these fields yourself. For more information, see System Fields.

Usage

In version 12.0 and later, use this object to track the votes that users have made on ideas. For more information on ideas, see "About Ideas" in the Salesforce online help.

WebLink

Represents a custom link to a URL or Scontrol.

Corresponds to an SObject4 in which the ObjectType="WebLink".

Supported Calls

Create(),Update(),Query(),Search(),Retrieve(),CreateObject(),DescribeSObjects()

Special Access Rules

- To create a custom link, the client application must be logged in with the "Customize Application" permission.
- Customer Portal users cannot access this object.

Field	Field Type	Field Properties	Description
Description	textarea	Create	Description of the custom link. Limit is 1,000 characters.
		Filter	
		Nillable	
		Update	
DisplayType	picklist	Create	Type of display: button, link, or mass-action button.
		Filter	
		Nillable	
		Restricted picklist	
		Update	
EncodingKey	picklist	Create	Required. Encoding of parameters on the URL link.
		Filter	
		Restricted picklist	
		Update	
HasMenubar	boolean	Create	Indicates whether the popup window shows a menu bar (true) or not (false).
		Defaulted on create	
		Filter	
		Update	

Field	Field Type	Field Properties	Description
HasScrollbars	boolean	Create Defaulted on create Filter	Indicates whether the popup window shows scroll bars (true) or not (false).
		Update	
HasToolbar	boolean	Create	Indicates whether the popup window shows browser toolbars
		Defaulted on create	(true) or not (false). Toolbars normally contain navigation buttons like Back, Forward, and Print.
		Filter	
		Update	
Height	int	Create	Height of the popup in pixels.
		Filter	
		Update	
		Nillable	
IsProtected	boolean	Create	Indicates whether the object is protected (true) or not
		Filter	(false). Protected components that have been installed in other organizations cannot be linked to or referenced by
		Update	components created in the subscriber organization. A developer can easily delete a protected component contain in a managed package in a future release of the package without worrying about failing installations. However, on a component is marked as unprotected and is released globally, the developer cannot delete it.
IsResizable	boolean	Create	Indicates whether users are allowed to resize the popup
		Defaulted on create	window (true) or not (Ialse).
		Filter	
		Update	
LinkType	picklist	Create	Required. Type of link (S-control or URL).
		Filter	
		Restricted picklist	
		Update	
MasterLabel	string	Create	Master label for the link. Limit is 240 characters. This display
		Filter	value is the internal label that is not translated.
		Update	
		Nillable	
Field	Field Type	Field Properties	Description
---------------------	------------	---------------------	-----------------------------------------------------------------------------------------------------------------------------------
Name	string	Create	Required. Name to display on page.
		Filter	
		Update	
		Nillable	
NamespacePrefix	string	Create	The namespace prefix assigned to this object when it was
		Filter	created as part of a managed package. Null if this object is not part of a managed package. Limit is 15 characters. For
		Nillable	more information about managed packages and namespace prefixes, see "About Managed Packages" in the Salesforce online help.
			This field cannot be accessed unless the logged-in user has the "Customize Application" permission.
OpenType	picklist	Create	Required. How the custom link opens when clicked in a
		Filter	browser—NewWindow, Sidebar, or NoSidebar.
		Restricted picklist	
		Update	
PageOrSObjectType	picklist	Create	Required. For standard objects, the name of the page on
		Filter	which to display the custom link. For custom objects, the
		Restricted picklist	name of the object.
Position	picklist	Create	Location on the screen where the popup should
		Filter	open—TopLeft, FullScreen, or None.
		Nillable	
		Restricted picklist	
		Update	
RequireRowSelection	boolean	Create	Indicates whether the custom link requires a row selection
		Defaulted on create	(true) or not (false).
		Filter	
		Nillable	
ScontrolId	reference	Create	ID of the custom s-control object (Scontrol) to link to. Can
		Filter	include Salesforce fields as tokens within the custom s-control object. For information on IDs see ID Field Type Label is
		Update	Custom S-Control ID.

Field	Field Type	Field Properties	Description
ShowsLocation	boolean	Create	Indicates whether the popup window shows the browser's
		Defaulted on create	address bar containing the URL (true) or not (false).
		Filter	
		Update	
ShowsStatus	boolean	Create	Show the status bar at the bottom of the browser.
		Defaulted on create	
		Filter	
		Update	
Url	string	Create	Required. URL of the page to link to. Can include Salesforce
		Nillable	fields as tokens within the URL. Limit: 1,024 Kb.
		Update	
Width	int	Create	Width of the popup in pixels.
		Filter	
		Nillable	
		Update	

Usage

Use this object to programmatically manage custom links, which allow client applications to integrate Salesforce data with external URLs, an organization's intranet, or other back-end office systems. A custom link can point to:

- An external URL, such as www.google.com or your company's intranet.
- A custom s-control, such as a Java applet or Active-X control.

Custom links can include Salesforce fields as tokens within the URL or custom s-control.

WebLinkLocalization

When the translation workbench is enabled for your organization, the WebLinkLocalization object provides the translation of the field label of a custom link to a URL or s-control. For information on the translation workbench, see the Salesforce online help.

Supported Calls

```
Create(),Update(),Query(),Retrieve(),CreateObject(),DescribeSObjects()
```

Special Access Rules

- Your organization must be using Professional, Enterprise, Developer, or Unlimited Edition and be enabled for the translation workbench.
- To view this object, you must have the "View Setup and Configuration" permission.

Fields

Field	Field Type	Field Properties	Description
LanguageLocaleKey picklist Create Filter The locale, which controls the in an application. String is 2-5 characters are always an ISO la Nillable or "en." If the value is further q string also has an underscore (_ code, for example "US" or "UK the United States is "en_US", a Canadian is "fr_CA." For a list of the languages that Salesforce online help topic "W support?"	The locale, which controls the language for labels displayed in an application. String is 2-5 characters long. The first two characters are always an ISO language code, for example "fr" or "en." If the value is further qualified by country, then the		
		Restricted picklist	string also has an underscore (_) and another ISO countr code, for example "US" or "UK. For example, the string for the United States is "en_US", and the string for French Canadian is "fr_CA."
			For a list of the languages that Salesforce supports, see the Salesforce online help topic "What languages does Salesforce support?"
WebLinkId	reference	Create	The ID of the WebLink that is being translated.
		Filter	
		Nillable	
Value	string	Create	The actual translated label of the custom link. Label is
		Filter	Iranslation.
		Nillable	
		Update	

Usage

Use this object to translate your custom links to URLs or s-controls into the different languages supported by Salesforce. Users with the translation workbench enabled can view custom link translations, but either the "Customize Application" or "Manage Translation" permission is required to Create() or Update() custom link translations.

Chapter 8

API COM Objects

The Office Toolkit provides access to the following COM objects:

Object	Description	
MergeRequest8	Represents a structure that is passed as a parameter to merge records of the same object type into one of the records.	
PickListValue4	Represents a picklist value. For internal use only.	
QueryResultSet4	presents a query result set returned from any of the following calls: Query(), trieve(), Search(), GetDeleted(), or GetUpdated(). After invoking one of ese calls, a client application will navigate the query result to inspect the results of the call.	
SBriefcaseType	Represents a briefcase. For internal use only.	
SError	Represents the success (NO_SF_ERROR) or error in an API call. Office Toolkit objects include an Error field that contains an array of SError objects. Client applications generally test for no error during conditional execution. For an example of error handling code, see Error Handling.	
SforceSession4	Represents a user session on the API, which is returned by after a successful Login() call.	
SforceSessionEvents4	Events raised by the SforceSession4 object.	
SObject4	Represents an object (see Standard and Custom Object Basics). The object type is specified in the ObjectType property for theSObject4 object. The EntityNames field in the SforceSession4 object contains a string array of the names of all Force.com objects available to the client application.	
Tab4	Provides information about the standard and custom tabs available to the logged-in user.	
TabSet4	Provides information about the standard and custom apps available to the logged-in user. An app is a group of tabs that works as a unit to provide application functionality. For example, two of the standard Salesforce apps are "Sales" and "Service and Support."	
TabSetCollection4	This object is a collection of the standard and custom apps in your organization. You can use the collection to get information about the apps available to the logged-in user.	

Object	Description
UserInfoResult8	Contains the personal information about the currently logged-in user. This information includes common profile data that your client application can use for display purposes, performing currency calculations, and so on.



Note: Certain properties are reserved for internal use only, even if they are visible in the Object Browser. These are listed in this document but are not documented.

ChildRelationship4

Represents a parent-child relationship. Describes which objects are related to this object and what is the type of relationship.

Properties

Name	Туре	Description	Read-Only?
CascadeDelete	Boolean	For internal use only.	Yes
ChildFieldName	String	For internal use only.	Yes
ChildSObjectName	String	For internal use only.	Yes
Error	SError	Error. See SError for details.	Yes
ErrorMessage	String	Error message text.	Yes
Tag	Variant	Provided as a convenience property for developers to use.	

Field4

Represents a field in an SObject4. The Item field in the SObject4 contains an array of Field4 objects. The Fields field in the SObject4 contains the names of all the available fields in the SObject4.

Property	Data Type	Description	Read-Only?
AutoNumber	Boolean	Indicates whether this is an autonumber field (true) or not (false). Analogous to a SQL IDENTITY type, autonumber fields are read only, non-createable text fields with a maximum length of 30 characters. Autonumber fields	Yes

Property	Data Type	Description	Read-Only?
		are used to provide a unique ID that is independent of the internal object ID (such as a purchase order number or invoice number). Autonumber fields are configured entirely in the Salesforce user interface. Use this attribute to determine whether a field is an autonumber field.	
ByteLength	Long	For variable-length fields (including binary fields), the maximum size of the field, in bytes.	Yes
CalculatedFormula	String	Formula that returns the calculated value for this field.	Yes
CaseSensitive	Boolean	Indicates whether the unique values for this field are case sensitive (true) or not (false).	Yes
ControllerName	String	Name of the picklist field that controls the values of this field.	Yes
Createable	Boolean	Indicates whether the field can be created (true) or not (false). If true, then this field value can be set in aCreate() call.	Yes
Custom	Boolean	Indicates whether the field is a custom field (true) or not (false).	Yes
DefaultOnCreate	Boolean	Indicates whether this field is defaulted when created (true) or not (false). If true, then Salesforce implicitly assigns a value for this field when the object is created, even if a value for this field is not passed in on the Create() call. For example, in the Opportunity object, the Probability field has this attribute because its value is derived from the Stage field. Similarly, the Owner has this attribute on most objects because its value is derived	Yes

Property	Data Type	Description	Read-Only?
		from the current user (if the Owner field is not specified).	
DefaultValueFormula	String	Formula that calculates the default value of the field.	Yes
DependentFields	Variant	The array of fields controlled by this field.	Yes
Digits	Long	For fields of type integer. Maximum number of digits. The API returns an error if an integer value exceeds the number of digits.	Yes
Error	SError	Error. See SError for details.	Yes
ErrorMessage	String	Error message text.	Yes
ExternalId	Boolean	Indicates whether this is an external ID field (true) or not (false).	Yes
Filterable	Boolean	Indicates whether the field is filterable (true) or not (false). If true, then this field can be specified in the WHERE clause of a query string in a Query () call.	Yes
HasData	Boolean	Indicates whether the field contains data (true) or not (false) that the client application populated either by setting the value or obtaining the value from the Salesforce data.	Yes
HtmlFormatted	Boolean	Indicates whether the field value is HTML-formatted (true) or not (false).	Yes
IsCalculated	Boolean	Indicates whether the field is a custom formula field (true) or not (false). Note that custom formula fields are always read-only.	Yes
IsDependentPicklist	Boolean	Indicates whether the field is a dependent picklist (true) or not (false).	Yes
IsMasterPicklist	Boolean	Indicates whether the field is the master of a dependent picklist (true) or not (false).	Yes

Property	Data Type	Description	Read-Only?
Label	String	Text label that is displayed next to the field in the Salesforce user interface. This label can be localized.	Yes
Length	Long	For string fields, the maximum size of the field in Unicode characters (not bytes).	Yes
Name	String	Field name used in API calls, such as Create() and Query().	Yes
NameField	Boolean	Indicates whether this field is a name field (true) or not (false). Used to identify the name field for standard objects (such as AccountName for an Account object) and custom objects. Limited to one per entity, except where FirstName and LastName fields are used (such as in the Contact object).	Yes
NamePointing	Boolean	Foreign key to the Name entity for aggregate queries.	Yes
Nillable	Boolean	Indicates whether the field is nillable (true) or not (false). A nillable field can have empty content. A non-nillable field must have a value in order for the object to be created or saved.	Yes
PickListValues	Variant	Array of PickListValue4 values representing the list of valid values for the picklist. Specified only if RestrictedPicklist is true.	Yes
Precision	Long	For fields of type double. Maximum number of digits that can be stored, including all numbers to the left and to the right of the decimal point (but excluding the decimal point character).	Yes
RelationshipName	String	Name of the foreign key relationship.	Yes

Property	Data Type	Description	Read-Only?
ReferenceTo	String[]	For fields that refer to other objects, this array indicates the objects of the referenced objects.	Yes
RestrictedPicklist	Boolean	Indicates whether the field is a restricted picklist (true) or not (false).	Yes
Scale	Long	For fields of type double. Number of digits to the right of the decimal point. The API silently truncates any extra digits to the right of the decimal point, but it returns a fault response if the number has too many digits to the left of the decimal point.	Yes
Sortable	Boolean	Indicates whether the field can be used to sort SOQL query results (true) or not (false).	Yes
Tag	Variant	Provided as a convenience property for developers to use as needed.	
Туре	String	SOAP data type. For the corresponding local application data type, see VariantType instead. For more information, see Office Toolkit and Proxies.	Yes
Unique	Boolean	Indicates whether values for this field must be unique (true) or not (false).	Yes
Updateable	Boolean	Indicates whether the field is updateable (true) or not (false). If true, then this field value can be set in a Update () call.	Yes
Value	Variant	Data value for this field. The data type is defined by the VariantType.	
VariantType	Long	Local application data type. The Long value represents the number associated with the enumerated variant type (for example VT_DATE=7, VT_BOOL=11, and so on).	Yes

Property	Data Type	Description	Read-Only?
		For the corresponding SOAP data type, see Type instead. For more information, see Office Toolkit and Proxies.	

Functions

Property	Returns	Description
GetDependentPicklistValues	Variant	Gets the array of values that are dependent on this field.
get_HasUserData	Boolean	Indicates whether the field value was set by the user or client application (true) or not (false).
PickListDefault	PickListValue4	For internal use only.
PickListValuesForRT		For internal use only.

LocalizationContext4

Represents a localization context. For internal use only.

Properties

Name	Туре	Description	Read-Only?
Error	SError	Error. See SError for details.	Yes
ErrorMessage	String	For internal use only.	Yes
Tag	Variant	For internal use only.	

Function	Return Type	Description
GetLabel	String	For internal use only.
GetLabelX	String	For internal use only.
LoadAndCheck	Boolean	For internal use only.
LoadLocal	BooleanFor internal use only.	For internal use only.

MergeRequest8

Represents a structure that is passed as a parameter to merge records of the same object type into one of the records.

Properties

Property	Data Type	Description	Read-Only
Master	SObject4	The SObject that is the master of the merge request. This is the record that the other records merge into.	No
RecordIdToMerge	Variant	ID of the record to be merged into the master record.	No

PickListValue4

Represents a picklist value. For internal use only.

Name	Туре	Description	Read-Only?
Active	Boolean	Indicates whether this item must be displayed (true) or not (false) in the drop-down list for the picklist field in the user interface.	Yes
Error	SError	Error. See SError for details.	Yes
ErrorMessage	String	Error message text.	Yes
IsDefault	Boolean	Indicates whether this item is the default item (true) in the picklist or not (false). Only one item in a picklist is designated as the default.	Yes
Label	String	Display name of this item in the picklist.	Yes
Tag	Variant	Provided as a convenience property for developers to use as needed.	
Value	String	Value of this item in the picklist.	Yes
ValidFor	String	Indicates which field is valid for the value property.	Yes

QueryResultSet4

Represents a query result set returned from any of the following calls: Query (), Retrieve (), Search (), GetDeleted (), or GetUpdated (). After invoking one of these calls, a client application will navigate the query result to inspect the results of the call.

Properties

Name	Туре	Description	Read-Only?
EntityType	String	Type of Force.com object that this QueryResultSet4 contains.	Yes
Error	SError	Error. See SError for details.	Yes
ErrorMessage	String	Error message text.	Yes
LastConfigChange	Variant	Last time the configuration changed.	Yes
Removed	Variant	Array of record IDs for records removed since the last call.	Yes
Size	Long	Total number of results for the query.	Yes
Tag	Variant	Provided as a convenience property for developers to use as needed. For example, you could use it to track the number of records that have been processed:qr.Tag = numberProcessed	
Token	String	Name of the token returned with the GetOfflineData call.	Yes

SBriefcaseType

Represents a briefcase. For internal use only.

Name	Value	Description
BRIEFCASE_NOT_SET	0	For internal use only.
BRIEFCASEANDACCOUNT	1	For internal use only.
BRIEFCASEANDACTIVITY	2	For internal use only.
BRIEFCASEANDOPPORTUNITY	3	For internal use only.

Name	Value	Description
BRIEFCASEONLY	4	For internal use only.

SError

Represents the success (NO_SF_ERROR) or error in an API call. Office Toolkit objects include an Error field that contains an array of SError objects. Client applications generally test for no error during conditional execution. For an example of error handling code, see Error Handling.

The following table describes each SError constant.

SError	Description
API_CURRENTLY_DISABLED	Because of a system problem, API functionality is temporarily unavailable.
API_DISABLED_FOR_ORG	The API has not been enabled for the organization. Contact salesforce.com to enable API access.
ASSIGNEE_TYPE_REQUIRED	Workflow tasks must have an assignee type field.
BATCH_CALL_ERROR	The batch call failed.
CANNOT_CASCADE_PRODUCT_ACTIVE	You cannot activate or deactivate this product without also being able to edit pricebooks.
CANNOT_DEACTIVATE_DIVISION	You cannot deactivate Divisions if an assignment rule references divisions or if a user's DefaultDivision field is not set to null.
CANNOT_INSERT_UPDATE_ACTIVATE_ENTITY	The user cannot insert, update, or activate this object.
CANNOT_REPARENT_RECORD	This record cannot be reparented.
CANNOT_RESOLVE_NAME	The name could not be resolved.
CANNOT_UPDATE_CONVERTED_LEAD	You cannot update a lead that has been converted.
CANT_DISABLE_CORP_CURRENCY	You cannot disable the corporate currency.
CANT_UNSET_CORP_CURRENCY	You cannot directly reset the corporate currency from the API. Instead, set a new corporate currency, which will clear previous settings.
CHILD_SHARE_FAILS_PARENT	You cannot change the ownership or sharing of the child record because the new owner or assignee does not have access to the parent record.
CIRCULAR_DEPENDENCY	There cannot be circular dependencies in certain situations, such as groups.
CUSTOM_ENTITY_OR_FIELD_LIMIT	The upper limit on custom objects or custom fields has been reached.
CUSTOM_FIELD_INDEX_LIMIT_EXCEEDED	For a given object, you cannot create more than the maximum allowed number of indexes.

SError	Description
DELETE_FAILED	This object is in use and cannot be deleted.
DEPENDENCY_EXISTS	There is a data dependency.
DUPLICATE_CASE_SOLUTION	The case solution relationship already exists.
DUPLICATE_DEVELOPER_NAME	You cannot have duplicate developer names on custom objects or custom fields.
DUPLICATE_MASTER_LABEL	You cannot have duplicate master names on custom objects or custom fields.
DUPLICATE_USERNAME	A duplicate username was specified in a call to insert or update a user.
DUPLICATE_VALUE	A duplicate value was specified.
EMAIL_TO_CASE_LIMIT_EXCEEDED	The daily converted email limit for the Email-to-Case feature has been exceeded.
EMAIL_TO_CASE_NOT_ENABLED	The Email-to-Case feature has not been enabled.
ENTITY_DESCRIPTION_LOAD_FAILURE	The database could not load the description for the entity.
ENTITY_DESCRIPTION_LOAD_FAILURE_ENTITY_NOT_FOUND	The database could not find the description for the entity.
ENTITY_FAILED_IFLASTMODIFIED_ON_UPDATE	The object failed the ifLastModifiedDate test on update.
ENTITY_IS_ARCHIVED	The object is archived.
ENTITY_IS_DELETED	The object is deleted; it cannot be deleted again or updated.
ENTITY_IS_LOCKED	The object has been locked.
ERROR_GETTING_LABELS	Error retrieving the labels.
EXCEEDED_ID_LIMIT_ON_RETRIEVE	Too many IDs were requested in a retrieve call.
EXCEEDED_LEAD_CONVERT_LIMIT	Too many IDs were sent to a ConvertLead() call.
EXCEEDED_MAX_SIZE_REQUEST	The size of the message sent to the API exceeded 50 MB.
EXCEEDED_MAX_TYPES_LIMIT	The number of object types to describe is too large.
EXCEEDED_QUOTA	The size limit for organization data storage was exceeded during a create call.
EXCEEDED_RATE_LIMIT	Concurrent API requests were sent by the client and the original request has been terminated.
FAILED_ACTIVATION	The contract or order failed to activate.
FAILED_TO_CREATE_DIRECTORY	Failed to create the directory.
FAILED_TO_GET_TEMPFOLDER	Failed to get the temporary directory.
FAILED_TO_OPEN_FILE	Failed to open the file.
FAILED_TO_READ_FILE	Failed to read the file.
FIELD_CUSTOM_VALIDATION_EXCEPTION	The data entered into the field has failed a custom validation formula.
FIELD_INTEGRITY_EXCEPTION	A field integrity error occurred.

SError	Description
FUNCTIONALITY_NOT_ENABLED	Functionality has been temporarily disabled. Other calls may continue to work.
HTTP_ERROR	An HTTP error occurred.
HTTP_REQUEST_TIMED_OUT	A request to the server resulted in no response.
IMAGE_TOO_LARGE	The image exceeds the maximum width, height, and file size.
INACTIVE_OWNER_OR_USER	The owner or user is inactive.
INSUFFICIENT_ACCESS	The user does not have sufficient access to perform the operation.
INSUFFICIENT_ACCESS_ON_CROSS_REFERENCE_ENTITY	An operation affects an object that is cross-referenced by the specified object, but the logged-in user does not have sufficient permissions on the cross-referenced object. For example, if the logged in user attempts to modify an account record, that user might not have permission to approve, reject, or reassign a ProcessInstanceWorkitem that is submitted after that action.
INSUFFICIENT_ACCESS_OR_READONLY	You do not have sufficient access rights for this record, or this record is read-only.
INVALID_ACCESS_LEVEL	Sharing rules must be less restrictive than the organization's default sharing level.
INVALID_APP_DATA_FOLDER	The application data folder is invalid.
INVALID_ARGUMENT_TYPE	An invalid argument type was specified.
INVALID_ASSIGNEE_TYPE	Assignee types must be one of the valid constant values.
INVALID_ASSIGNMENT_RULE	The assignment rule is invalid or is not defined for the organization.
INVALID_BATCH_OPERATION	The batch operation is invalid.
INVALID_BATCH_SIZE	The query options have an invalid batch size value.
INVALID_CLIENT	The client is invalid.
INVALID_CLIENT_FOR_LOCALIZATION	The client cannot retrieve localization labels.
INVALID_CREDIT_CARD_INFO	The credit card information is invalid.
INVALID_CROSS_REFERENCE_KEY	The cross-reference key is invalid.
INVALID_CROSS_REFERENCE_TYPE_FOR_FIELD	The cross-reference type is not valid for that field.
INVALID_CURRENCY_CONV_RATE	You must specify a positive, non-zero value for the currency conversion rate.
INVALID_CURRENCY_ISO	The three-letter currency abbreviation (ISO code) is invalid.
INVALID_EMAIL_ADDRESS	An invalid email address was specified.
INVALID_EMPTY_KEY_OWNER	You cannot set an empty key as the owner.
INVALID_FIELD	The specified field name is invalid.
INVALID_FIELD_FOR_INSERT_UPDATE	An invalid field name was specified in an insert or update call.

SError	Description
INVALID_FILTER_ACTION	You cannot associate certain actions or rules with certain objects, such assigning alerts to tasks.
INVALID_ID_FIELD	An invalid ID field, cross-reference, or owner ID field was specified.
INVALID_INET_ADDRESS	A specified Inet address is not valid.
INVALID_LABEL_FILE	The label file is invalid.
INVALID_LABEL_PATH	The label path is invalid.
INVALID_LINEITEM_CLONE_STATE	To be cloned, pricebooks and pricebook entries must be active.
INVALID_LOCAL_ID	The local ID is invalid.
INVALID_LOGIN	The login credentials are invalid.
INVALID_MANIFEST_FILE	The manifest file is invalid.
INVALID_NEW_PASSWORD	The new password does not conform with the password policies of the organization.
INVALID_OPERATION	The client application tried to submit an object that is already in process as part of workflow approval processing.
INVALID_OPERATION_WITH_EXPIRED_PASSWORD	Due to password expiration, a valid password must be set using SetPassword () before the call can be invoked.
INVALID_OPERATOR	An invalid operation was specified, such as sfdc.list or sfdc.get.
INVALID_OR_NULL_FOR_RESTRICTED_PICKLIST	An invalid or null value was specified for a restricted picklist field.
INVALID_QUERY_FILTER_OPERATOR	An invalid operator was used in the query filter clause, at least for that field.
INVALID_QUERY_LOCATOR	An invalid queryLocator parameter was specified in a queryMore call.
INVALID_QUERY_SCOPE	The specified search scope is invalid.
INVALID_REPLICATION_DATE	The date for replication is out of the allowed range, such as before the organization was created.
INVALID_SEARCH	The search request has invalid syntax or grammar.
INVALID_SEARCH_SCOPE	The specified search scope is invalid.
INVALID_SERVLET_RESPONSE	The servlet has returned an invalid response.
INVALID_SESSION_ID	The specified SessionId is invalid or has expired. Log in again to start a new session.
INVALID_SOAP_HEADER	There is an error in the SOAP headers. If you are migrating from an earlier version of the API, be advised that the SaveOptions header cannot be used with API version 6.0 or later. Note that you cannot set AssignmentRuleHeader.
INVALID_SSO_GATEWAY_URL	The URL provided to configure the Single Sign-On gateway was not valid.

SError	Description
INVALID_STATUS	An invalid status was specified.
INVALID_TYPE	An invalid type of record was specified.
INVALID_TYPE_FOR_OPERATION	An invalid type was specified for the operation.
INVALID_TYPE_ON_FIELD_IN_RECORD	An invalid type was specified on a field in the record.
LAST_MODIFIED_SINCE_TOO_OLD	The value for Last Modified Since is too far in the past.
LICENSE_LIMIT_EXCEEDED	You have exceeded the maximum allowed number of licenses for your organization.
LOCALE_NOT_SUPPORTED	The locale is not supported.
LOGIN_DURING_RESTRICTED_DOMAIN	The user is not allowed to log in from this IP address.
LOGIN_DURING_RESTRICTED_TIME	The user is not allowed to log in during this time period.
MALFORMED_ID	An ID must be either 15 or 18 characters long.
MALFORMED_QUERY	An invalid query string was specified. For example, the query string was longer than 10,000 characters.
MALFORMED_SEARCH	An invalid search string was specified. For example, the search string was longer than 10,000 characters.
MAX_ACTIONS_PER_RULE_EXCEEDED	You have exceeded the maximum allowed number of workflow actions per rule.
MAX_ACTIVE_RULES_EXCEEDED	You have exceeded the maximum allowed number of active rules per object for your organization.
MAX_TASK_DESCRIPTION_EXCEEEDED	The task description is too long.
MAXIMUM_CCEMAILS_EXCEEDED	You have exceeded the maximum allowed number of CC: emails in workflow alerts.
MAXIMUM_DASHBOARD_COMPONENTS_EXCEEDED	The size of the dashboard component is too large.
MAXIMUM_SIZE_OF_ATTACHMENT	This size of the attachment is too large.
MAXIMUM_SIZE_OF_DOCUMENT	The size of the document is too large.
MISSING_ARGUMENT	A required argument was not specified.
NO_SF_ERROR	Status message indicating that no error was returned.
NONUNIQUE_SHIPPING_ADDRESS	You cannot insert a reduction order item if the original order shipping address is different from the shipping address of other items in the reduction order.
NUMBER_OUTSIDE_VALID_RANGE	The number is not in the allowed range.
OPERATION_TOO_LARGE	The query has returned too many results. If certain queries are run by a user without the "View All Data" permission and many records are returned, those queries would require sharing rule checking. An example of this would be queries that are run on objects, such as Task, that use a polymorphic foreign key. Such queries return this exception because the operation requires too many resources. To correct, add filters to the query

SError	Description
	to narrow the scope, or use filters such as date ranges to break the query up into a series of smaller queries.
ORG_LOCKED	The organization has been locked. You must contactsalesforce.com to unlock the organization.
PASSWORD_EXPIRED	The password has expired.
PASSWORD_LOCKOUT	The user has exceeded the allowed number of login attempts. The user must contact his or her administrator to regain login access.
PRIVATE_CONTACT_ON_ASSET	You cannot have a private contact on an asset.
QUERY_TIMEOUT	The query has timed out. For more information, see Salesforce Object Query Language (SOQL)
REQUIRED_FIELD_MISSING	A required field was not specified.
SERVER_UNAVAILABLE	A server that is necessary for this call is currently unavailable. Some requests may still work.
SFDB_FAILED_CREATE_TABLE	The database failed to create a table.
SFDB_INTERNAL_ERROR	An internal database error has occurred.
SFDB_PARSE_ERROR	A parsing error occurred in the database.
SFDB_QUERY_ERROR	A query failed.
SFDB_SYNC_IN_PROGRESS	A database synchronization is currently in progress.
SFDB_USER_VALID_BUT_NOT_SYNCED	You must sync once to use the local database.
SFLOOKUP_UKNOWN_ENTITY	Unknown lookup entity.
SFORCE_INTERNAL_ERROR	There is an internal error. Please see the error status message for more details.
SHARE_NEEDED_FOR_CHILD_OWNER	The parent share cannot be deleted; the child owner needs it.
SSO_SERVICE_DOWN	The service was unavailable, and an authentication call to the organization's specified Single Sign-On server failed.
STANDARD_PRICE_NOT_DEFINED	Custom prices cannot be defined without corresponding standard prices.
STORAGE_LIMIT_EXCEEDED	You have exceeded the maximum allowed data storage.
TEXT_DATA_OUTSIDE_SUPPORTED_CHARSET	The text data is outside of the supported character set.
TOO_MANY_ENUM_VALUE	A request failed because too many values were passed in for a multi-select picklist. You can select a maximum of 100 values for a multi-select picklist.
TRANSFER_REQUIRES_READ	You cannot assign the record to the specified user because the user does not have read permission.
TRIAL_EXPIRED	The trial period for the organization has expired. A representative from the organization must contactsalesforce.com to re-enable the organization.

SError	Description
UNKNOWN_EXCEPTION	An unspecified error occurred.
UNRESOLVED_OFFLINE_FK	Unresolved database foreign key.
UNSPECIFIED_EMAIL_ADDRESS	The user does not have an email address.
UNSUPPORTED_API_VERSION	The file represents an unsupported version of the API.
UNSUPPORTED_CLIENT	This version of the client is no longer supported.

SforceSession4

Represents a user session on the API, which is returned by after a successful Login () call.

Name	Туре	Description	Read-Only?
CurrencySymbol	String	Currency symbol to use for displaying currency values. Applicable only when OrganizationMultiCurrency is false.	Yes
CurrentServerTime	Variant	System timestamp of the API. The value is refreshed from the system each time this property is accessed by a client application.	Yes
CurrentUserInfo	UserInfoResult	Information in the current user's user record, including role ID, email, user ID, and so on.	Yes
EarliestStartDateAvailable	Variant	Earliest date for a GetDeleted() or GetUpdated() call start time. You can set this after a failed attempt.	Yes
EntityNames	Variant	String array containing the names of all Force.com objects that are available to the client application. Each array element represents a separate object type.	Yes
Error	SError	Error. See SError for details.	Yes
ErrorMessage	String	Error message text.	Yes

Name	Туре	Description	Read-Only?
HTTPConnectTimeout	Long	HTTP connection timeout, in milliseconds.	
HTTPReceiveTimeout	Long	HTTP receive timeout, in milliseconds.	
IsLoggedIn	Boolean	Specifies whether the current session is valid.	
LatestDateAvailable	Variant	Latest date for a GetDeleted() or GetUpdated() call end time. You can set this after a failed attempt.	Yes
OnAckOfflineDataFinished	Variant	Specifies the script callback function for an ack offline data finished callback.	
OnCreateFinished	Variant	Specifies the script callback function for a Create() call.	
OnEndElement	Variant	For internal use only.	
OnGetDeletedFinished	Variant	Specifies the script callback function for a GetDeleted() call.	
OnGetOfflineDataFinished	Variant	Specifies the script callback function for a get offline data finished callback.	
OnGetUpdatedFinished	Variant	Specifies the script callback function for a GetUpdated() call.	
OnQueryFinished	Variant	Specifies the script callback function for a Query () call.	
OnRefreshFinished	Variant	Specifies the script callback function for a Refresh() call.	
OnRetrieveFinished	Variant	Specifies the script callback function for a Retrieve () call.	
OnSearchFinished	Variant	Specifies the script callback function for a Search () call.	
OnStartElement	Variant	For internal use only.	
OnSynchFinished	Variant	For internal use only.	
OnSynchStatus	Variant	For internal use only.	
OnUpdateFinished	Variant	Specifies the script callback function for an Update() call.	

Name	Туре	Description	Read-Only?
OrganizationId	ID	Organization ID. Allows third-party tools to uniquely identify individual organizations in Salesforce, which is useful for retrieving billing or organization-wide setup information. For more information about IDs, see ID Field Type.	Yes
OrganizationMultiCurrency	Boolean	Indicates whether the user's organization uses multiple currencies (true) or not (false).	Yes
OrganizationName	String	Name of the user's organization or company.	Yes
PasswordExpired	Boolean	Indicates whether the password has expired (true) or not (false). Read-only.	Yes
Route	Boolean	For internal use only.	
ServerUrl	String	URL returned by the API after a successful login. Could differ from the URL provided in the Login () call. Used internally as the endpoint for subsequent API calls.	Yes
SessionId	String	Current session ID.	
Tag	Variant	Provided as a convenience property for developers to use as needed.	
UserDefaultCurrencyIsoCode	String	Default currency ISO code. Applicable only when OrganizationMultiCurrency is true. When the logged-in user creates any objects that have a currency ISO code, the system uses this currency ISO code if it is not explicitly specified in the Create() call.	Yes
UserEmail	String	User's email address.	Yes
UserFullName	String	User's full name.	Yes
UserID	ID	User ID. For more information about IDs, see ID Field Type	Yes

Name	Туре	Description	Read-Only?
UserLocale	String	User's locale (language and country).	Yes
UserTimeZone	String	User's time zone.	Yes

Function	Returns	Description
AckOfflineData		For internal use only.
CancelSync		For internal use only.
ConfigOfflineDB		For internal use only.
Create()		Creates one or more SObject4 objects in the database.
CreateObject()	SObject4	Creates a new instance of the specified object.
Delete		Deletes one or more SObject4 objects in the database.
DescribeSObjects()	Variant	Describes a number of object types at once.
DoLookupSearch	String	Performs a lookup.
DoTZConversions	Boolean	Turns time zone conversions on and off.
GetOfflineData	QueryResultSet4	For internal use only.
GetTabSets	TabSetCollection4	Returns an enumerable TabSetCollection4 with the current TabSets.
EmptyOfflineStore		For internal use only.
GetDeleted()	QueryResultSet4	Retrieves objects that have been deleted since the specified time for the specified object.
GetLocalizationContext	LocalizationContext4	For internal use only.
GetUpdated()	LocalizationContext4	Retrieves objects that have been updated since the specified time for the specified object.
GetUserInfo		For internal use only. User information is returned in the SforceSession4 object.
Login()	Boolean	Logs into the server with the specified credentials.
MakeHttpRequest	String	For internal use only.
Mobile	Boolean	For internal use only. Returns local mobile setting.

Function	Returns	Description
Parse		For internal use only.
Query()	QueryResultSet4	Executes a query against the specified object and returns data that matches the specified criteria. For details, see Salesforce Object Query Language (SOQL).
QueryAll	QueryResultSet4	Queries all objects, including those in the Recycle Bin.
QueryOfflineDB	String	For internal use only.
Refresh()		Refreshes the specified SObject4 objects with the latest data from the database.
ResolveDBErrors	Boolean	For internal use only.
Retrieve()	QueryResultSet4	Retrieves one or more objects based on the specified object ID(s). The fieldList can be a wildcard (*). For more information about IDs, see ID Field Type
Search()	QueryResultSet4	Searches for objects based on keywords. For details, see Salesforce Object Search Language (SOSL).
SetBriefcase	SBriefcaseType	For internal use only.
SetCookie		For internal use only.
SetServerURL()		Sets the Force.com Web service URL endpoint prior to login.
SetSOAPHeader()		Sets SOAP header options for configuring the batch size for Query () calls and lead assignment rules.
SynchOfflineDB		For internal use only.
Undelete()		Undeletes a record.
Update()		Updates one or more SObject4 objects in the database.
UseCache		Uses cached information to describe objects and layouts.

Events

Event	Description
CreateFinished	Triggered after a Create() call has completed.
DBSynchFinished	For internal use only.
DBSynchStatus	For internal use only.

Event	Description
EndElement	For internal use only.
GetDeletedFinished	Triggered after a GetDeleted() call has completed.
GetUpdatedFinished	Triggered after a $GetUpdated()$ call has completed.
LoginFinished	Triggered after a Login () call has completed.
QueryFinished	Triggered after a Query () call has completed.
RefreshFinished	Triggered after a Refresh() call has completed.
RetrieveFinished	Triggered after a Retrieve() call has completed.
SearchFinished	Triggered after a Search () call has completed.
StartElement	For internal use only.
UpdateFinished	Triggered after an Update () call has completed.
UndeleteFinished	Triggered after an Undelete () call has completed.

SforceSessionEvents4

Events raised by the SforceSession4 object.

Events

Event	Description
AckOfflineDataFinished	Triggered after an AckOfflineData event completes.
CreateFinished	Triggered after a Create() call completes.
GetDeletedFinished	Triggered after a GetDeleted() call completes.
GetOfflineDataFinished	Triggered after a GetOfflineData event completes.
GetUpdatedFinished	Triggered after a GetUpdated() call completes.
QueryFinished	Triggered after a Query() call completes.
RefreshFinished	Triggered after a Refresh() call completes.
RetrieveFinished	Triggered after a Retrieve () call completes.
SearchFinished	Triggered after a Search () call completes.
UndeleteFinished	Triggered after an Undelete () call completes.
UpdateFinished	Triggered after an Update () call completes.

SObject4

Represents an object (see Standard and Custom Object Basics). The object type is specified in the ObjectType property for theSObject4 object. The EntityNames field in the SforceSession4 object contains a string array of the names of all Force.com objects available to the client application.

Property	Data Type	Description	Read-Only
Activateable	Boolean	Reserved for future use.	Yes
BulkEditLayout	String	For internal use only.	Yes
ChildRelationships	Variant	Describes which objects are related to this object and the type of relationship.	Yes
Createable	Boolean	Indicates whether the SObject4 can be created via the Create() call (true) or not (false).	Yes
Custom	Boolean	Indicates whether the SObject4 is a custom object (true) or not (false).	Yes
DefaultRecordTypeID	String	For internal use only.	Yes
Deletable	Boolean	Indicates whether the SObject4 can be deleted.	Yes
Error	SError	Error. See SError for details.	Yes
ErrorMessage	String	Error message text.	Yes
Fields	String	String array containing the names of all fields in the SObject4 that are available to the client application. Each array element represents a separate field in the SObject4.	Yes
HasServerData	Boolean	Indicates whether this SObject4 already exists in the database (true) or not (false). This property indicates whether the SObject4 was loaded from a Retrieve(), Refresh(), Query() or Search() calls, (true) or from a blank object (false). For example, this property returns false immediately after a CreateObject() call.	Yes
Item	Field4	Array of Field4 objects. Each element in the array represents a field (name-value pair) in the SObject4.	Yes
KeyPrefix	String	Three character code prefix in the object ID, which specifies the object type. For example, Account objects have a prefix of " 001", Opportunity objects have a prefix of " 006". For object types that do not have a stable or predictable prefix, this field is blank. Thus, client applications that rely on these codes should use this way of determining object types to ensure forward compatibility.	Yes

Property	Data Type	Description	Read-Only
Label	String	Returns the label text for a renamed tab (for example, "Patient" in a medical vertical) in the user interface, if applicable, or the object name, if not.	Yes
Layoutable	Boolean	For internal use only.	Yes
LayoutXML	String	For internal use only.	Yes
Mergeable	Boolean	Indicates whether the property can be merged with other properties of its type (true) or not (false).	Yes
ObjectType	String	Returns a String representation of the object type (such as "Account").	Yes
PluralLabel	String	Label text for an object that represents the plural version of an object name, for example, "Accounts".	Yes
Queryable	Boolean	Returns whether the SObject4 can be queried via the Query() call (true) or not (false).	Yes
RecordTypeSelectRequired	Boolean	For internal use only.	Yes
Replicateable	Boolean	Returns whether the SObject4 can be replicated via the GetUpdated() and GetDeleted() calls (true) or not (false).	Yes
Retrievable	Boolean	Indicates whether the property is retrievable.	Yes
Searchable	Boolean	Returns whether the SObject4 can be searched via the Search() call (true) or not (false).	Yes
Tag	Variant	Provided as a convenience property for developers.	Yes
Undeletable	Boolean	Reserved for future use.	Yes
Updateable	Boolean	Returns whether the SObject4 can be updated via the Update() call (true) or not (false).	Yes
UrlDetail	String	URL to the view the detail page (read-only) for this object. Compare with urlEdit, which is read-write. If a stable URL is not available for this object, this field is returned empty. Client applications use this URL to redirect to, or access, the Salesforce user interface for standard and custom objects. To provide flexibility including future enhancements, returned urlDetail values are dynamic. Ensure that client applications are forward compatible by using this capability.	Yes
UrlEdit	String	URL to the edit page for this object. For example, the urlEdit field for the account object returns https://nal.salesforce.com/{ID}/e. Substituting the {ID} field for the current object ID will return the Salesforce user interface's edit page for that object. Compare with urlDetail, which is read-only. Client applications can use this URL to redirect to, or access, the standard Salesforce user interface for standard and custom objects. To provide flexibility and allow for future enhancements, returned	Yes

Property	Data Type	Description	Read-Only
		urlEdit values are dynamic. To ensure that client applications are forward compatible, it is recommended that they use this capability where possible. Note that, for objects for which a stable URL API is not available, this field is returned empty.	
UrlNew	String	URL to the "create new" page for this object type. Client applications can use this URL to redirect to, or access, the standard Salesforce user interface for standard and custom objects. To provide flexibility and allow for future enhancements, returned urlNew values are dynamic. To ensure that client applications are forward compatible, it is recommended that they use this capability where possible. Note that, for objects for which a stable URL API is not available, this field is returned empty.	Yes

Function	Returns	Description
Clone	SObject4	Clones the current SObject4. Syntax:SObject4.Clone() As SObject4
Create	SObject4	Creates the current SObject4 in the database. Syntax: SObject4.Create()
Delete		Deletes the current SObject4 in the database. Syntax:SObject4.Delete()
GetJoinResults	SObject4	Gets join results by relationship name.
GetLayoutXML	String	Returns the layout XML for the specified layout ID. Internal use only. Syntax: SObject4.GetLayoutXML(id As String) As String
Item	Field4	Returns an Field4 object with the specified field name or field index. Syntax: SObject4.Item(key) As Field4
RecordTypeIdAvailable	Boolean	Returns whether the record type, specified by record type ID, is available to the current user (true) or not (false). Syntax: Sbject4.RecordTypeIdAvailable(_MIDL_0014 As String) As Boolean
RecordTypeLayoutId	String	Returns the layout ID for the specified record type ID. Syntax:

Function	Returns	Description
		Subject4.RecordTypeLayoutId(_MIDL_0013 As String) As String
Refresh		Refreshes the data in this SObject4 instance with the most recent data in the database. Syntax: SObject4.Refresh()
Update		Updates the current SObject4 in the database. Syntax: SObject4.Update()

Tab4

Provides information about the standard and custom tabs available to the logged-in user.

Properties

Property	Data Type	Description	Read-Only
Custom	Boolean	Specifies whether this is a custom tab (true). or a standard tab (false).	Yes
IconUrl	String	The URL for the 32 x 32 pixel icon for the tab. This icon appears next to the heading at the top of most pages.	Yes
Label	String	The display label for this tab.	Yes
MiniIconUrl	String	The URL for the 16 x 16 pixel icon that represents the tab. This icon appears in related lists and other locations.	Yes
SObjectName	String	The name of the SObject4 that is primarily displayed on this tab (for tabs that display a particular SObject). For a list of objects, see Standard Objects.	Yes
Url	String	The fully qualified URL for viewing this tab.	Yes
UIClass	String	The css class of the user interface used to display this tab.	Yes

TabSet4

Provides information about the standard and custom apps available to the logged-in user. An app is a group of tabs that works as a unit to provide application functionality. For example, two of the standard Salesforce apps are "Sales" and "Service and Support."

Properties

Property	Data Type	Description	Read-Only
Label	String	Display label for the app. This value changes when the app is renamed in the Salesforce user interface. See the Salesforce online help for more information.	Yes
LogoUrl	String	A fully qualified URL to the logo image associated with the app.	Yes
Namespace	String	If this is a custom app, and a set of tabs in the custom app was installed as part of a managed package, the value of this attribute is the developer namespace prefix that the creator of the package chose when the Developer Edition organization was enabled to allow publishing of a managed package. This attribute identifies elements of a Force.com AppExchange package.	Yes
Selected	Boolean	If true, then this app is the user's currently selected app.	Yes
Tabs	Tab4	The array of tabs that are displayed for this app.	Yes

TabSetCollection4

This object is a collection of the standard and custom apps in your organization. You can use the collection to get information about the apps available to the logged-in user.

Properties

Property	Data Type	Description	Read-Only
Count	Long	Number of tabsets in this collection.	Yes

Function	Returns	Description
Item	Variant	Name of the tabset collection. This method allows you to access a particular tabset through an index value.
_NewEnum	Unknown	Returns a copy of the current collection object. This is a standard method in all collection objects and returns an enumerator object that supports the EnumVARIANT interface. This interface is a collection of Variants that allows users to enumerate collections of objects without knowing specific types

Function	Returns	Description
		of elements in the collection. In other words, this method allows you to iterate over TabSets.

UserInfoResult8

Contains the personal information about the currently logged-in user. This information includes common profile data that your client application can use for display purposes, performing currency calculations, and so on.

This object applies only to the username under which your client application has logged in.

Name	Туре	Description	Read-Only?
AccessibilityMode	Boolean	Indicates whether user interface modifications for the visually impaired are on (true) or off (false). The modifications facilitate the use of screen readers such as JAWS.	Yes
CurrencySymbol	String	Currency symbol to use for displaying currency values. Applicable only when organizationMultiCurrency is false.	Yes
LicenseType	String	The category of user license. Each license type is associated with one or more user licenses. Each license is associated with one or more profiles.	Yes
OrganizationId	String	ID of the organization. Allows third-party tools to uniquely identify individual organizations in Salesforce, which is useful for retrieving billing or organization-wide setup information.	Yes
OrganizationMultiCurrency	Boolean	Indicates whether the user's organization uses multiple currencies (true) or not (false).	Yes
OrganizationName	String	Name of the user's organization or company.	Yes
ProfileId	String	ID of the profile associated with the role currently assigned to the user.	Yes
RoleId	String	ID of the role currently assigned to the user.	Yes
UserDefaultCurrencyIsoCode	String	Default currency ISO code. Applicable only when organizationMultiCurrency is true. When the logged-in user creates any objects that have a currency ISO code, the API uses this currency ISO code if it is not explicitly specified in the Create() call.	Yes
UserEmail	String	User's email address.	Yes
UserFullName	String	User's full name.	Yes
UserId	ID	User's record ID.	Yes

Name	Туре	Description	Read-Only?
UserLanguage	String	User's language, which controls the language for labels displayed in an application. String is 2-5 characters long. The first two characters are always an ISO language code, for example "fr" or "en." If the value is further qualified by country, then the string also has an underscore (_) and another ISO country code, for example "US" or "UK. For example, the string for the United States is "en_US", and the string for French Canadian is "fr_CA."	Yes
		For a list of the languages that Salesforce supports, see the Salesforce online help topic "What languages does Salesforce support?"	
UserLocale	String	User's locale, which controls the formatting of dates and choice of symbols for currency. The first two characters are always an ISO language code, for example "fr" or "en." If the value is further qualified by country, then the string also has an underscore (_) and another ISO country code, for example "US" or "UK. For example, the string for the United States is "en_US", and the string for French Canadian is "fr_CA."	Yes
UserName	String	User's login name.	Yes
UserTimeZone	String	User's time zone.	Yes
UserUiSkin	String	Available in API version 7.0 and later. Returns the value Theme2 if the user is using the newer user interface theme of the online application, labeled "Salesforce." Returns Theme1 if the user is using the older user interface theme, labeled "Salesforce Classic." In the online application, this look and feel setting is configurable at Setup Customize > User Interface	Yes

Event	Description
Equals	Used to test the equality of two UserInfoResult8 objects.

Chapter 9

Core Calls

The following table lists supported calls in the API in alphabetical order, and provides a brief description for each. Click a call name to see syntax, usage, and more information for that call.

Call	Description
ConvertLead()	Converts a Lead into an Account, Contact, or (optionally) an Opportunity.
Create()	Adds one or more new individual objects to your organization's data.
CreateObject()	Creates an instance of an SObject4.
GetDeleted()	Retrieves the IDs of individual objects of the specified object that have been deleted since the specified time. For information on IDs, see ID Field Type.
GetUpdated()	Retrieves the IDs of individual objects of the specified object that have been updated since the specified time. For information on IDs, see ID Field Type.
Login()	Logs in to the login server and starts a client session.
Query()	Executes a query against the specified object and returns data that matches the specified criteria.
Refresh()	Retrieves the latest data from the database and updates the local object instance in a client application with that data.
Retrieve()	Retrieves one or more objects based on the specified object IDs.
Search()	Executes a text search in your organization's data.
SetServerURL()	Sets the Force.com API URL endpoint prior to login.
SetSOAPHeader()	Sets SOAP header options for configuring the batch size for $Query()$ calls and lead assignment rules.
Undelete()	Undelete records.
Update()	Updates one or more existing objects in your organization's data.

ConvertLead()

Converts a Lead into an Account, Contact, or (optionally) an Opportunity.

Syntax

```
LeadConvertResult[] = binding.convertLead(leadConverts LeadConvert[]);
```

Usage

Use ConvertLead() to convert a Lead into an Account and Contact, as well as (optionally) an Opportunity. To convert a Lead, your client application must be logged in with the "Convert Leads" permission and the "Edit" permission on leads, as well as "Create" and "Edit" on the Account, Contact, and Opportunity objects.

This call provides an easy way to convert the information in a qualified lead to a new or updated account, contact, and opportunity. Your organization can set its own guidelines for determining when a lead is qualified, but typically, a lead can be converted as soon as it becomes a real opportunity that you want to forecast.

If data is merged into existing account and contact objects, then only empty fields in the target object are overwritten—existing data (including IDs) are not overwritten. The only exception to this is if your client application sets overwritteLeadSource to true, in which case the LeadSource field in the target Contact object will be overwritten with the contents of the LeadSource field in the source Lead object.

When converting leads, consider the following rules and guidelines:

Field Mappings

The system automatically maps standard lead fields to standard account, contact, and opportunity fields. For custom lead fields, your Salesforce administrator can specify how they map to custom account, contact, and opportunity fields.

Record Types

If the organization uses record types, the default record type of the new owner is assigned to records created during lead conversion. For more information about record types, see the Salesforce online help.

Picklist Values

The system assigns the default picklist values for the account, contact, and opportunity when mapping any standard lead picklist fields that are blank. If your organization uses record types, blank values are replaced with the default picklist values of the new record owner.

Basic Steps for Converting Leads

Converting leads involves the following basic steps:

- 1. The client application determines the IDs of any lead(s) to be converted.
- 2. Optionally, the client application determines the IDs of any account(s) to merge the lead into. The client application can use SOSL or SOQL to search for accounts that match the lead name, as in the following example:

select id, name from account where name='CompanyNameOfLeadBeingMerged'

3. Optionally, the client application determines the IDs of contact(s) to merge the lead into. The client application can use SOSL or SOQL to search for contacts that match the lead contact name, as in the following example:

select id, name from contact where firstName='FirstName' and lastName='LastName' and accountId = '001...'

- 4. Optionally, the client application determines whether opportunities should be created from the leads.
- 5. The client application queries the LeadSource table to obtain all of the possible converted status options (SELECT ... FROM LeadStatus WHERE IsConverted='1'), and then selects a value for the Converted Status.

- 6. The client application calls ConvertLead().
- 7. The client application iterates through the returned result(s) and examine each LeadConvertResult object to determine whether conversion succeeded for each lead.
- 8. As an optional best practice, the client application creates tasks in which the WhoId is the ContactId and, if an opportunity is created, the WhatId is the OpportunityId.
- **9.** Optionally, when converting leads owned by a queue, the owner must be specified. This is because accounts and contacts cannot be owned by a queue. Even if you are specifying an existing account or contact, you must still specify an owner.

Sample Code—Java

```
private Boolean convertLead (String leadId, String contactId,
String accountId, boolean overWriteLeadSource, boolean doNotCreateOpportunity,
String opportunityName, String convertedStatus, boolean sendEmailToOwner)
 LeadConvert leadConvert = new LeadConvert();
 leadConvert.setLeadId(new ID(leadId));
 leadConvert.setContactId(new ID(contactId));
 leadConvert.setAccountId(new ID(accountId));
 leadConvert.setOverwriteLeadSource(overWriteLeadSource);
 leadConvert.setDoNotCreateOpportunity(doNotCreateOpportunity);
 leadConvert.setOpportunityName(opportunityName);
 leadConvert.setConvertedStatus(convertedStatus);
 leadConvert.setSendNotificationEmail(sendEmailToOwner);
 LeadConvertResult[] lcr = null;
 try {
    lcr = binding.convertLead(new LeadConvert[] {leadConvert});
    for (int i=0; i<lcr.length; i++) {</pre>
      if (lcr[i].isSuccess()) {
        System.out.println("Conversion succeeded.\n");
        LeadConvertResult result = lcr[i];
        System.out.println("The new contact id is: " + result.getContactId());
      } else {
        System.out.println("The conversion failed because: " +
lcr[i].getErrors(0).getMessage());
   }
  } catch (UnexpectedErrorFault e) {
   System.out.println("Unexpected error encountered:\n\n" + e.getExceptionMessage());
   return false;
  } catch (RemoteException e) {
   System.out.println("Remote exception encountered:\n\n" + e.getMessage());
    return false;
 return true;
}
```

Sample Code—C#

```
private bool convertLead(string leadId, string contactId,
    string accountId, bool overWriteLeadSource,
    bool doNotCreateOpportunity, string opportunityName,
    string convertedStatus, bool sendEmailToOwner)
{
    sforce.LeadConvert leadConvert = new sforce.LeadConvert();
    leadConvert.leadId = leadId;
    leadConvert.contactId = contactId;
    leadConvert.accountId = accountId;
    leadConvert.overwriteLeadSource = overWriteLeadSource;
    leadConvert.ovportunity = doNotCreateOpportunity;
    leadConvert.convertedStatus = convertedStatus;
    leadConvert.sendNotificationEmail = sendEmailToOwner;
```

ConvertLead()

```
sforce.LeadConvertResult[] lcr = null;
try
{
   lcr = binding.convertLead(new sforce.LeadConvert[] {leadConvert});
   for (int i=0;i<lcr.Length;i++)</pre>
      if (lcr[i].success)
     {
         Console.WriteLine("Conversion succeeded.\n");
         sforce.LeadConvertResult result = lcr[i];
         Console.WriteLine("The new contact id is: " + result.contactId);
      }
      else
       Console.WriteLine("The conversion failed because: " + lcr[i].errors[0].message);
      }
}
catch (Exception e)
   Console.WriteLine("Unexpected error encountered:\n\n" + e.Message);
   return false;
return true;
```

LeadConvert Arguments

This call accepts an array of LeadConvert objects (100 maximum). A LeadConvert object contains the following properties.

Name	Туре	Description
accountId	ID	ID of the Account into which the lead will be merged. Required only when updating an existing account, including person accounts. If no accountID is specified, then the API creates a new account. To create a new account, the client application must be logged in with sufficient access rights. To merge a lead into an existing account, the client application must be logged in with read/write access to the specified account. The account name and other existing data are not overwritten. For information on IDs, see ID Field Type.
contactId ID	ID	ID of the Contact into which the lead will be merged (this contact must be associated with the specified accountId, and an accountId must be specified). Required only when updating an existing contact.
		Important: If you are converting a lead into a person account, do not specify the contactId or an error will result. Specify only the accountId of the person account.
		If no contactID is specified, then the API creates a new contact that is implicitly associated with the Account. To create a new contact, the client application must be logged in with sufficient access rights. To merge a lead into an existing contact, the client application must be logged in with read/write access to the specified contact. The contact name and other existing data are not overwritten (unless overwriteLeadSource is set to true, in which case only the

Name	Туре	Description
		LeadSource field is overwritten). For information on IDs, see ID Field Type.
convertedStatus	string	Valid LeadStatus value for a converted lead. Required. To obtain the list of possible values, the client application queries the LeadStatus object, as in:
		Select Id, MasterLabel from LeadStatus where IsConverted=true
doNotCreateOpportunity	boolean	Specifies whether to create an Opportunity during lead conversion (false, the default) or not (true). Set this flag to true only if you do not want to create an opportunity from the lead. An opportunity is created by default.
leadId	ID	ID of the Lead to convert. Required. For information on IDs, see ID Field Type.
opportunityName	string	Name of the opportunity to create. If no name is specified, then this value defaults to the company name of the lead. The maximum length of this field is 80 characters. If doNotCreateOpportunity argument is true, then no Opportunity is created and this field must be left blank; otherwise, an error is returned.
overwriteLeadSource	boolean	Specifies whether to overwrite the LeadSource field on the target Contact object with the contents of the LeadSource field in the source Lead object (true), or not (false, the default). To set this field to true, the client application must specify a contactId for the target contact.
ownerId	ID	Specifies the ID of the person to own any newly created account, contact, and opportunity. If the client application does not specify this value, then the owner of the new object will be the owner of the lead. Not applicable when merging with existing objects—if an ownerId is specified, the API does not overwrite the ownerId field in an existing account or contact. For information on IDs, see ID Field Type.
sendNotificationEmail	boolean	Specifies whether to send a notification email to the owner specified in the ownerId (true) or not (false, the default).

Response

LeadConvertResult[]

LeadConvertResult

This call returns an array of LeadConvertResult objects. Each element in the LeadConvertResult array corresponds to the LeadConvert[] array passed as the leadConverts parameter in the ConvertLead() call. For example, the object returned in the first index in the LeadConvertResult array matches the object specified in the first index of the LeadConvert[] array. A LeadConvertResult object has the following properties:
Name	Туре	Description
accountId	ID	ID of the new Account (if a new account was specified) or the ID of the account specified when ConvertLead() was invoked.
contactId	ID	ID of the new Contact (if a new contact was specified) or the ID of the contact specified when ConvertLead() was invoked. For information on IDs, see ID Field Type.
leadId	ID	ID of the converted Lead. For information on IDs, see ID Field Type.
opportunityId	ID	ID of the new Opportunity, if one was created when $\texttt{ConvertLead}()$ was invoked. For information on IDs, see ID Field Type.
success	boolean	Indicates whether the ConvertLead() call succeeded (true) or not (false) for this object.

Create()

Adds one or more new individual objects to your organization's data.

Syntax

```
Create(SObject4[], async As Boolean);
```

Usage

Use Create () to add one or more individual objects, such as an Account or Contact, to your organization's information. The Create () call is analogous to the INSERT statement in SQL.

When creating objects, consider the following rules and guidelines.

Permissions

Your client application must be logged in with sufficient access rights to create individual objects within the specified object. For more information, see Factors that Affect Data Access.

Special Handling

Certain objects—and certain fields within those objects—require special handling or permissions. For example, you might also need permissions to access the object's parent object. Before you attempt to Create() a particular object, be sure to read its description in the Standard Objects.

Createable Fields

Only objects where Createable is true can be created via the Create () call.

Automatically Maintained Fields

The API generates unique values for ID fields automatically. For Create(), you cannot explicitly specify an ID value in the SObject4. The SError[] property (in the SforceSession4 object) contains the ID of each object that was successfully created. For information on IDs, see ID Field Type.

The API populates certain fields automatically, such as CreatedDate, CreatedById, LastModifiedDate, LastModifiedById, and SystemModstamp. You cannot explicitly specify these values.

Required Fields

For required fields that do not have a preconfigured default value, you must supply a value. For more information, see Required Fields.

Default Values

For some objects, some fields have a default value, such as OwnerID. If you do not specify a value for such fields, the API populates the fields with the default value. For example, if you do not override OwnerID, then the API populates this field with the user ID associated with the user as whom your client application is logged in.

- For required fields that do not have a preconfigured default value, you must supply a value.
- For all other fields in the object, if you do not explicitly specify a value, then its value is null (VT EMPTY).

Referential Integrity

Your client application must conform to the rules of referential integrity. For example, if you are creating an object that is the child of a parent object, you must supply the foreign key information that links the child to the parent. For example, when creating a CaseComment, you must supply the valid caseID for the parent Case, and that parent Case must exist in the database.

Valid Data 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 specified for your programming language and development tool (your development tool will handle 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 name field is entered as " ABC Company ", then the value is stored in the database as "ABC Company".

Assignment Rules

When creating new Account (accounts fire Territory Management assignment rules), Case, or Lead objects, your client application can set options in the SOAP header (see SetSOAPHeader ()) to have the case or lead automatically assigned to one or more users based on assignment rules configured in the Salesforce user interface.

Maximum Number of Objects Created

Your client application can add up to 200 individual objects in a single Create () call. If a create request exceeds 200 objects, then the entire operation fails.

Relationships

Although you can use relationships in Query () calls, you cannot create objects of different types even if they have a parent-child relationship, for example accounts and contacts.

Create()

You can use external ID fields as a foreign key, allowing you to create or update records in a single step instead of querying a record to get the ID first. To do this, specify the foreign key name and the external ID field value. For example:

Basic Steps for Creating Objects

Creating objects involves the following basic steps:

- 1. Instantiate one or more individual objects from the session. For each object, populate its fields with the data that you want to add.
- 2. Construct an SObject4[] array and populate that array with the objects that you want to create. All objects must be of the same type.
- 3. Call Create(), passing in the SObject4[] array.
- 4. Process the results in the SObject4[] property to verify whether the objects have been successfully created.

Sample Code—VBA Synchronous

```
Public Function CreateAccount() As SObject4
```

On Error GoTo handleError

'create using batch call

Dim so(0) As SObject4

Set so(0) = g_sfApi.CreateObject("account")

so(0) ("name") = "Test Array account"

'call batch method

g sfApi.Create so, False

'call batch refresh

g sfApi.Refresh so, False

Exit Function

handleError:

MsgBox g sfApi.ErrorMessage

End Function

Arguments

Name	Туре	Description
SObject4	SObject4[]	Array of one or more SObject4 arrays to Create(). Limit: 200 SObject4 values.
async	boolean	Specifies whether this invocation should be processed asynchronously (true) or not (false).

Event

CreateFinished

CreateObject()

Creates a new instance of an SObject4 with the specified object type.

Syntax

```
SObject4
= CreateObject(objectName As String);
```

Usage

Use CreateObject() to create a new instance of an SObject4 with the specified object type. Client applications generally use this factory method to instantiate an object instance, populate the newly-instantiated object with data values, and then call the SObject4. Create method to create the object in the database.

Sample Code—VBA

```
'Create a blank object
Public Function CreateAccount() As SObject4
On Error GoTo handleError
'all entities are of type SObject4
   Dim account As SObject4
'create a blank object
   Set account = g_sfApi.CreateObject("account")
    account("name") = "Sample"
'commit the object
   account.Create
'refresh the object to get server set values
    account.Refresh
'set as the function return value
    Set CreateAccount = account
handleError:
  MsgBox g_sfApi.ErrorMessage
```

End Function

Arguments

Name	Туре	Description
objectName	string	Name of the object to create (such as
		"Account").

Event

None

GetDeleted()

Retrieves the list of individual objects that have been deleted within the given timespan for the specified object.

Syntax

QueryResultSet4[] = GetDeleted(EntityType As String, startDate, endDate, async As Boolean);

Usage

Use GetDeleted() for data replication applications to retrieve a list of object instances that have been deleted from your organization's data within the specified timespan. The GetDeleted() call retrieves a QueryResultSet4 object that contains an array of DeletedRecord objects containing the ID of each deleted object and the date/time (Coordinated Universal Time (UTC) time zone) on which it was deleted. Be sure to read Data Replication before using GetDeleted() in your client applications. (For information on IDs, see ID Field Type.)

As of release 8.0, the GetDeleted () call respects the user's sharing model.

Rules and Guidelines

When replicating deleted objects, consider the following rules and guidelines:

- The specified startDate must chronologically precede the specified endDate value. The specified startDate cannot be the same value as, or later than, the specified endDate value. Otherwise, the API returns an INVALID REPLICATION DATE error.
- Records are returned only if the user has access to them.
- Results are returned for no more than 30 days previous to the day the call is executed (or earlier if an administrator has purged the recycle bin). If the purge has been performed before your GetDeleted() call is executed, an INVALID_REPLICATION_DATE error is returned.
- Deleted records are written to a delete log, which GetDeleted () accesses. A background process that runs every two hours purges records that have been in an organization's delete log for more than two hours if the number of records is above a certain limit. Starting with the oldest records, the process purges delete log entries until the delete log is back below the limit. This is done to protect Salesforce from performance issues related to massive delete logs. The limit is calculated using this formula:

For example, an organization with 1,000 licenses could have up to 5,000,000 (five million) records in the delete log before any purging took place. If purging has been performed before your GetDeleted() call is executed, an INVALID REPLICATION DATE error is returned. If you get this exception, you should do a full pull of the table.

- If you delete a large numbers of records, your data replication should run more frequently than every two hours to ensure all records are returned by GetDeleted().
- Client applications typically poll for changed data periodically. For important polling considerations, see Polling for Changes.
- Certain objects cannot be replicated via the API. To replicate an object via the GetDeleted() call, its object must be configured as replicateable (rReplicateable is true in the SObject4). To determine whether a given object can be replicated, your client application can invoke the call on the object and inspect its property.
- Development tools differ in the way that they handle time data. Some development tools report the local time, while others report only the Coordinated Universal Time (UTC) time. To determine how your development tool handles time values, refer to its documentation.

Basic Steps for Replicating Deleted Objects

You can replicate deleted objects using the following basic steps for each object:

- 1. Optionally, determine whether the structure of the object has changed since the last replication request, as described in Checking for Structural Changes in the Object.
- 2. Call GetDeleted (), passing in the object and the relevant time span during which object were deleted.
- 3. In the QueryResultSet4 object, iterate through the returned array of DeletedRecord objects containing the ID of each deleted object and the date on which it was deleted (Coordinated Universal Time (UTC) time zone).
- 4. Take the appropriate action on the local data to remove the deleted objects or flag as deleted.
- 5. Optionally, save the request time span for future reference. You should save the value of .

A client application likely performs other tasks associated with data replication operations. For example, if an opportunity is closed, a client application might run a new revenue report. Similarly, if a task is completed, the process might log this in another system.

Sample Code—VBA

```
Function GetDeleted()
On Error GoTo handleError
Dim qr As QueryResultSet4
Dim v As Variant
Dim s As SObject4
Dim startDate As Date
startDate = Now - 29 'start date cannot be more than 30 days ago
Dim endDate As Date 'today
endDate = Now
Set qr = g_sfApi.GetDeleted("account", startDate, endDate, False)
For Each v In qr
'loop through the results
'cast to an SObject4 to see more helpful debug info
Set s = v
```

Next v

Exit Function

handleError:

MsgBox Err.Description

End Function

Arguments

Name	Туре	Description
sObjectTypeEntityType	string	Object type. The specified value must be a valid object for your organization. See SObject4.
startDate	dateTime	Starting date/time (Coordinated Universal Time (UTC)—not local— timezone) of the timespan for which to retrieve the data. The API ignores the seconds portion of the specified dateTime value (for example, 12:30:15 is interpreted as 12:30:00 UTC).
endDate	dateTime	Ending date/time (Coordinated Universal Time (UTC)—not local— timezone) of the timespan for which to retrieve the data. The API ignores the seconds portion of the specified dateTime value (for example, 12:35:15 is interpreted as 12:35:00 UTC).
async	boolean	Specifies whether this invocation should be processed asynchronously (true) or not (false).

Limits

There are record limits on the result :

- If your GetDeleted() call returns more than 600,000 records and the user is a system administrator, an exception EXCEEDED_ID_LIMIT is returned.
- If your GetDeleted() call returns more than 20,000 records and the user is not a system administrator, an exception OPERATION_TOO_LARGE is returned. Note that this error is returned when more than 20,000 records across the organization have been deleted, not just the records viewable by the user.

You can correct the error by choosing start and end dates that are closer together.

Event

GetDeletedFinished

GetUpdated()

Retrieves the list of individual objects that have been updated (added or changed) within the given timespan for the specified object.

Syntax

QueryResultSet4[] = GetUpdated(EntityType As String, startDate, endDate, async As Boolean);

Usage

Use GetUpdated () for data replication applications to retrieve a set of IDs for objects of the specified object that have been created or updated within the specified timespan. The GetUpdated () call retrieves an array of QueryResultSet4 objects containing the ID of each created or updated object and the date/time (Coordinated Universal Time (UTC) time zone) on which it was created or updated, respectively. Be sure to read Data Replication before using GetUpdated() in your client application.

Note: The GetUpdated() call retrieves the IDs only for objects to which the logged-in user has access.

Rules and Guidelines

When replicating created and updated objects, consider the following rules and guidelines:

- The specified startDate must chronologically precede the specified endDate value. The specified startDate cannot be the same value as, or later than, the specified endDate value. Otherwise, the API returns an INVALID REPLICATION DATE error.
- Results are returned for no more than 30 days previous to the day the call is executed.
- Client applications typically poll for changed data periodically. For important polling considerations, see Polling for Changes.
- Your client application can replicate any objects to which it has sufficient permissions. For example, to replicate all data for your organization, your client application must be logged in with "View All Data" access rights to the specified object. Similarly, the objects must be within your sharing rules. For more information, see Factors that Affect Data Access.
- Certain objects cannot be replicated via the API. To replicate an object via the GetUpdated() call, its object must be configured as replicateable (Replicateable is true in the SObject4).
- Certain objects cannot be deleted, such as Group, User, Contract, or Product2 objects. However, if instances of these objects are no longer visible in the Salesforce user interface, they may have been rendered inactive so that only users with administrative access can see them. To determine whether a missing object instance has been made inactive, your client application can call GetUpdated() and check the object's active flag.
- Development tools differ in the way that they handle time data. Some development tools report the local time, while others report only the Coordinated Universal Time (UTC) time. To determine how your development tool handles time values, refer to its documentation.

Basic Steps for Replicating Updated Objects

Replicating objects involves the following basic steps for each object that you want to replicate:

- 1. Optionally, the client application determines whether the structure of the object has changed since the last replication request, as described in Checking for Structural Changes in the Object.
- 2. Call GetUpdated (), passing in the object and timespan for which to retrieve data.
- 3. Iterate through the returned array of IDs. For each ID element in the array, call Retrieve() to obtain the latest information you want from the associated object. Your client application must then take the appropriate action on the local data, such as inserting new rows or updating existing ones with the latest information.
- 4. Optionally, the client application saves the request timestamp for future reference.

A client application likely performs other tasks associated with data replication operations. For example, if an opportunity were to become closed, a client application might run a new revenue report. Similarly, if a task were completed, the process might log this somehow in another system.

Sample Code—VBA

```
Function GetUpdated()
On Error GoTo handleError
    Dim qr As QueryResultSet4
    Dim v As Variant
    Dim s As SObject4
    Dim startDate As Date
    startDate = Now - 29 'start date cannot be more than 30 days ago
   Dim endDate As Date 'today
    endDate = Now
    Set qr = g_sfApi.GetUpdated("account", startDate, endDate, False)
    For Each v In qr
'loop through the results
'cast to a sobject4 to see more helpful debug info
        Set s = v
   Next v
    Exit Function
handleError:
   MsgBox Err.Description
```

End Function

Arguments

Name	Туре	Description
sObjectTypeEntityType	string	Object type. The specified value must be a valid object for your organization. For a list of standard objects, see Standard Objects.
startDate	dateTime	Starting date/time (Coordinated Universal Time (UTC) time zone—not local— timezone) of the timespan for which to retrieve the data. The API ignores the seconds portion of the specified dateTime value (for example, 12:30:15 is interpreted as 12:30:00 UTC).
endDate	dateTime	Ending date/time (Coordinated Universal Time (UTC) time zone—not local— timezone) of the timespan for which to retrieve the data. The API ignores the seconds portion of the specified dateTime value (for example, 12:35:15 is interpreted as 12:35:00 UTC).

Name	Туре	Description
async	boolean	Specifies whether this invocation should be processed asynchronously (true) or not (false).

Event

GetUpdatedFinished

Login()

Logs in to the login server and starts a client session.

Syntax

Boolean = Login(username As String, password As String);

Usage

Use the Login () call to log in to the login server and start a client session.

When a client application invokes the Login () call, it passes in a username and password as user credentials. Upon invocation, the Force.com API authenticates the credentials for the session, the user ID associated with the logged-in username, and a URL that points to the Force.com API to use in all subsequent API calls.

Salesforce checks the IP address from which the client application is logging in, and blocks logins from unknown IP addresses. For a blocked login via the API, Salesforce returns a login fault. Then, the user must add their security token to the end of their password in order to log in. A security token is an automatically-generated key from Salesforce. For example, if a user's password is mypassword, and their security token is XXXXXXXXX, then the user must enter mypasswordXXXXXXXXX to log in. Users can obtain their security token by changing their password or resetting their security token via the Salesforce user interface. When a user changes their password or resets their security token, Salesforce sends a new security token to the email address on the user's Salesforce record. The security token is valid until a user resets their security token, changes their password, or has their password reset. When the security token is invalid, the user must repeat the login process to log in. To avoid this, the administrator can make sure the client's IP address is added to the organization's list of trusted IP addresses. For more information, see Security Token.

Development tools differ in the way you specify session headers and server URLs. For more information, see the documentation for your particular development tool.

Enterprise and Partner Endpoints

In version 11.1 of the API and earlier, client applications built with the partner WSDL can send requests to the enterprise endpoint and enterprise WSDL applications can send requests to the partner endpoint. Beginning with version 12.0, this is not supported.

Session Expiration

Client applications do not need to explicitly log out to end a session. Sessions expire automatically after a predetermined length of inactivity, which can be configured in Salesforce by clicking **Setup** > **Security Controls**. The default is 120 minutes (two hours).

Sample Code—VBA

```
Public Function SampleLogin(UserName, Password)
'setup for exception type error handling
On Error GoTo handleError
'create a session object
   Set g_sfApi = New SForceOfficeToolkitLib3.SForceSession4
'make a login call
   g_sfApi.Login UserName, Password
   Exit Function
handleError:
'look at the exception message
   MsgBox Err.Description
'look at the message in the session
   MsgBox g_sfApi.ErrorMessage
End Function
```

Arguments

Name	Туре	Description
username	string	Login username.
password	string	Login password associated with the specified username.

Event

LoginFinished

Query()

Executes a query against the specified object and returns data that matches the specified criteria.

Syntax

```
QueryResultSet4[] = Query(qString As String, async As
Boolean);
```

Usage

Use the Query() call to retrieve data from an object. When a client application invokes the Query() call, it passes in a query expression that specifies the object to query, the fields to retrieve, and any conditions that determine whether a given object qualifies. For an extensive discussion about the syntax and rules used for queries, see Salesforce Object Query Language (SOQL).

Upon invocation, the API executes the query against the specified object, caches the results of the query on the API, and returns a query response object to the client application. The client application can then use methods on the query response object to iterate through rows in the query response and retrieve information.

Your client application must be logged in with sufficient access rights to query individual objects within the specified object and to query the fields in the specified field list. For more information, see Factors that Affect Data Access.

Certain objects cannot be queried via the API. To query an object via the Query () call, its object must be configured as queryable (query access is available in the SObject4). To determine whether an object can be queried, your client application can invoke the DescribeSObjects () call on the object and inspect its property.

You can also filter on the isArchived field to find only the archived objects. You cannot use Query (), it automatically filters out all records where isArchived is set to true. You can insert, update, or delete archived records.

Queries that take longer than two minutes to process will be timed out. For timed out queries, the API returns an API fault element of SError. If a timeout occurs, refactor your query to return or scan a smaller amount of data.

When querying for fields of type Base64 (see base64), the query response object returns only one record at a time. You cannot alter this by changing the batch size of the Query () call. Note that client applications are responsible for the conversion of Base64 values between binary data and the Strings representing the Base64 data. For example, the following function can handle the conversion of values:

```
Function Base64EncodeBinary(inData)
Base64EncodeBinary = Base64Encode(BinaryToString(inData))
```

End Function



Note: For multicurrency organizations, special handling is required when querying currency fields containing values in different currencies. For example, if a client application is querying PricebookEntry objects based on values in the UnitPrice field, and if the UnitPrice amounts are expressed in different currencies, then the query logic must handle this case correctly. For example, if the query is trying to retrieve the product codes of all products with a unit price greater than or equal to \$10 USD, the query expression might look something like this:

```
SELECT Product2Id,ProductCode,UnitPrice FROM PricebookEntry
WHERE (UnitPrice >= 10 and CurrencyIsoCode='USD')
OR (UnitPrice >= 5.47 and CurrencyIsoCode='GBP')
OR (UnitPrice >= 8.19 and CurrencyIsoCode='EUR')
```

Sample Code—VBA: Query—Synchronous Example

```
Public Function Query()
Dim qr As QueryResultSet4
Dim v As Variant
Dim s As SObject4
Set qr = g_sfApi.Query("select * from task", False)
```

```
For Each v In qr
'loop through the results
'cast to a sobject4 to see more helpful debug info
        Set s = v
'use the object
        s("Name") = "Query"
        's.Update
        Next v
End Function
```

Sample Code—VBA: Query—Asynchronous Example

```
Public Function AsyncQuery()
'run a query
    g_sfApi.Query "select * from task", True
End Function
'Asynchronous query callback
Public Sub g sfApi QueryFinished(qr As QueryResultSet4)
    Dim v As Variant
    Dim s As SObject4
'check session for errors
    If g sfApi.Error <> NO SF ERROR Then
'query has failed
        Debug.Print g sfApi.ErrorMessage
        Exit Sub
   End If
'look at query result attributes
'The size is the TOTAL size of the result
    Debug.Print "Size of result set:" & qr.Size
   For Each v In qr
'loop through the results
'cast to an SObject4 to see more helpful debug info
        Set s = v
'use the object
'NOTE you cannot call async methods until this method returns
```

's.Update Next v

End Sub

Arguments

Name	Туре	Description
queryString	string	Query string that specifies the object to query, the fields to return, and any conditions for including a specific object in the query. For more information, see Salesforce Object Query Language (SOQL).
async	Boolean	Specifies whether this invocation should be processed asynchronously (true) or not (false).

Event

QueryFinished

Salesforce Object Query Language (SOQL)

Use the Salesforce Object Query Language (SOQL) to construct simple but powerful query strings for the queryString parameter in the Query () call. Similar to the SELECT command in Structured Query Language (SQL), SOQL allows you to specify the source object (such as Account), a list of fields to retrieve, and conditions for selecting rows in the source object.



Note: SOQL does not support all advanced features of the SQL SELECT command. For example, you cannot use SOQL to perform arbitrary join operations, use wildcards in field lists, or use calculation expressions.

Use the following topics to understand the components of a SOQL SELECT, their syntax and usage:

- SOQL Typographical Conventions
- SELECT
- Changing the Batch Size in Queries
- toLabel()
- Querying Currency Fields in Multicurrency Organizations
- Date Formats and Date Literals
- null
- Filtering on Boolean Fields
- Relationship Queries
- Querying Multi-Select Picklists
- Syndication Feed SOQL and Mapping Syntax

SOQL Typographical Conventions

Topics about SOQL use the following typographical conventions:

Convention	Description
SELECT Name FROM Account	In an example, Courier font indicates items that you should type as shown. In a syntax statement, Courier font also indicates items that you should type as shown, except for question marks and square brackets.

Convention	Description
SELECT fieldname FROM objectname	In an example or syntax statement, italics represent variables. You supply the actual value.
?	In a syntax statement, the question mark indicates the element preceding it is optional. You may omit the element or include one.
WHERE [conditionexpression]	In a syntax statement, square brackets surround an element that may be repeated up to the limits for that element. You may omit the element, or include one or more of them.
SELECT Name FROM Account	In some examples, particular elements are highlighted with bold if they are of particular interest in the text before or after the example.

Alias Notation

You can use alias notation in SELECT queries:

SELECT count() from Contact c, c.Account a WHERE a.name = 'MyriadPubs'

To establish the alias, first identify the object, in this example a contact, and then specify the alias, in this case "c." For the rest of the SELECT statement, you can use the alias in place of the object or field name.

SELECT

The SOQL SELECT command uses the following syntax:

```
SELECT fieldList FROM objectType [WHERE conditionExpression]
[ORDER BY]
LIMIT ?
```

Syntax	Description
fieldList subquery?	 Specifies a list of one or more fields, separated by commas, that you want to retrieve from the specified <i>object</i>. The bold elements in the following examples are <i>fieldlists</i>: SELECT Id, Name, BillingCity FROM Account SELECT count() FROM Contact SELECT Contact.Firstname, Contact.Account.Name FROM Contact
	Specify the wildcard (*) to return all fields for the object. Because using the wildcard might return a very large result set that could slow client application performance, use it only when necessary. You must specify valid field names and must have read-level permissions to each specified field. The <i>fieldList</i> defines the ordering of fields in the query results. The order of the fields in the array is alphabetic by API field name when traversed from 0 to n.
	fieldList can include a subquery if the query will traverse a relationship. For example: SELECT Account.Name, (SELECT Contact.LastName FROM Account.Contacts) FROM Account
	Note: See Relationship Queries for more information about the syntax for parent-to-child and child-to-parent relationships represented by the last two examples in the list above.

Syntax	Description
	The fieldlist can also be count () or be wrapped in toLabel().
objectType	Specifies the type of object that you want to Query (). You must specify a valid object and must have read-level permissions to that object. For a list of valid objects, see Standard Objects.
conditionExpression	If WHERE is specified, determines which rows and values in the specified object $(objectType)$ to filter against. If unspecified, the Query () retrieves all the rows in the object that are visible to the user. See WHERE for the appropriate syntax.



Note: SOQL statements cannot exceed 10,000 characters. For SOQL statements that exceed this maximum length, the API returns a SError of MALFORMED_QUERY; no result rows are returned.

Condition Expression (WHERE Clause)

The conditionExpression, the WHERE clause, uses the following syntax:

fieldExpression logicalOperator fieldExpression2 ?

The condition expressions in SOQL SELECT statements appear in bold in these examples:

- SELECT Name FROM Account WHERE Name like 'A%'
- SELECT Id FROM Contact WHERE Name like 'A%' AND MailingCity='California'
- SELECT Name FROM Account WHERE CreatedDate > 2006-11-16T10:00:00-08:00

You can use date or datetime values, or date literals. The format for date and dateTime fields are different.

• You can use parentheses to define the order in which *fieldExpressions* are evaluated. For example, the following expression is true if fieldExpression1 is true and either fieldExpression2 or fieldExpression3 are true:

fieldExpression1 AND (fieldExpression2 OR fieldExpression3)

• However, the following expression is true if either fieldExpression3 is true or both fieldExpression1 and fieldExpression2 are true.

(fieldExpression1 AND fieldExpression2)OR fieldExpression3

• Client applications must specify parentheses when nesting operators. However, multiple operators of the same type do not need to be nested.



Note: The WHERE clause behaves in two different ways, depending on the version, when handling null values in a parent field for a relationship query. In a WHERE clause that checks for a value in a parent field, if the parent does not exist, the record is returned in Version 13.0 and later, but not returned in versions before 13.0.

SELECT Id FROM Case WHERE Contact.Lastname = null

Case record Id values are returned in version 13.0 and later, but are not returned in versions before 13.0. For more information, see Lookup Relationships and Outer Joins.

See *fieldExpression* for the syntax of *fieldExpression*. See Logical Operators for the valid logical operators.

fieldExpression

fieldExpression uses the following syntax:

```
fieldName comparisonOperator value
```

where:

Syntax	Description
fieldName	The name of a field in the specified object. Use of single or double quotes around the name will result in an error. You must have at least read-level permissions to the field. It can be any field except a long text area field, encrypted data field, or base64-encoded field. It does not need to be a field in the <i>fieldList</i> .
comparisonOperator	See Comparison Operators for a list of valid operators.
value	A value used to compare with the value in <i>fieldName</i> . You must supply a value whose data type matches the field type of the specified field. You must supply a native value—other field names or calculations are not permitted. For date values, use the formatting listed in Date Formats and Date Literals. If quotes are required (for example, they are not for dates and numbers), use single quotes. Double quotes result in an error.

Comparison Operators

The following table lists the *comparisonOperator* values that are used in *fieldExpression* syntax. Note that comparisons on strings are case-insensitive.

Operator	Name	Description
=	Equals	Expression is true if the value in the specified <i>fieldName</i> equals the specified <i>value</i> in the expression. String comparisons using the equals operator are case-insensitive.
! =	Not equals	Expression is true if the value in the specified <i>fieldName</i> does not equal the specified <i>value</i> .
<	Less than	Expression is true if the value in the specified <i>fieldName</i> is less than the specified <i>value</i> .
<=	Less or equal	Expression is true if the value in the specified fieldName is less than, or equals, the specified value.
>	Greater than	Expression is true if the value in the specified <i>fieldName</i> is greater than the specified <i>value</i> .
>=	Greater or equal	Expression is true if the value in the specified fieldName is greater than or equal to the specified value.
LIKE	Like	 Expression is true if the value in the specified fieldName matches the characters of the text string in the specified value. The LIKE operator in SOQL and SOSL is similar to the SAME operator in SQL; it provides a mechanism for matching partial text strings and includes support for wildcards. The % and _ wildcards are supported for the LIKE operator. The % wildcard matches zero or more characters. The _ wildcard matches exactly one character. The text string in the specified value must be enclosed in single quotes.

Operator	Name	Description
		 The LIKE operator is supported for string fields only (see string). The LIKE operator performs a case-insensitive match, unlike the case-sensitive matching in SQL. The LIKE operator in SOQL and SOSL does not currently support escaping of special characters % or You should not use the backslash character in a search (except to escape a character), as it is reserved.
		For example, the following query matches Appleton, Apple, and Bappl , but not Appl:
		SELECT AccountId, FirstName, lastname FROM Contact WHERE lastname LIKE 'appl_%'
IN	IN	If the value equals any one of the specified values in a WHERE clause. For example: SELECT Name FROM ACCOUNT WHERE BillingState IN ('California', 'New York')
		Note that the values for IN must be in parentheses. String values must be surrounded by single quotes.
		IN and NOT IN can also be used for semi-joins and anti-joins when searching on ID fields. For more information, see Semi-Joins with IN and Anti-Joins with NOT IN.
NOT IN	NOT IN	If the value does not equal any of the specified values in a WHERE clause. For example:
		SELECT Name FROM ACCOUNT WHERE BillingState NOT IN ('California', 'New York')
		Note that the values for NOT IN must be in parentheses, and string values must be surrounded by single quotes.
		There is also a logical operator NOT.
INCLUDES EXCLUDES		Applies only to multi-select picklists. See Querying Multi-Select Picklists.

Semi-Joins with ${\tt IN}$ and Anti-Joins with ${\tt NOT}~{\tt IN}$

You can query values in a field where another field on the same object has a specified set of values, using IN. For example:

SELECT Name FROM ACCOUNT WHERE BillingState IN ('California', 'New York')

In addition, you can create more complex semi-joins and anti-joins using IN and NOT IN to query ID fields.

Basic Semi-Join: Use in a WHERE Clause

You can include a semi-join in a WHERE clause. For example, the following query returns account IDs if an associated opportunity is lost:

```
SELECT Id, Name
FROM Account
WHERE Id IN
(SELECT AccountId FROM Opportunity WHERE StageName = 'Closed Lost')
```

Notice that the subquery returns a single field of the same type as the field to which it is compared. A full list of restrictions that prevent unnecessary processing is provided at the end of this section.

Basic Anti-Join: Use in a WHERE Clause

The following example returns account IDs for all accounts that do not have any open opportunities:

```
SELECT Id
FROM Account
WHERE Id NOT IN (SELECT AccountId FROM Opportunity WHERE IsClosed = false)
```

Multiple Semi-Joins or Anti-Joins

You can combine semi-join or anti-join clauses in a query. For example, the following query returns account IDs that have open opportunities if the last name of the contact associated with the account is like the last name "Apple":

```
SELECT Id, Name
FROM Account
WHERE Id IN (
   SELECT AccountId FROM Contact WHERE LastName LIKE 'apple%'
)
AND Id IN (
   SELECT AccountId FROM Opportunity WHERE isClosed = false)
```

You can use at most two subqueries in a single semi-join or anti-join query. Multiple semi-joins and anti-join queries are also subject to existing limits on subqueries per query. For more information, see the subquery limits described in Understanding Relationship Query Limitations.

Semi-Joins or Anti-Joins Evaluating Relationship Queries

You can create a semi-join or anti-join that evaluates a relationship query in a SELECT clause. For example, the following query returns opportunity IDs and their related line items if the opportunity's line item total value is more than \$10,000:

```
SELECT Id, (SELECT Id from OpportunityLineItems)
FROM Opportunity
WHERE Id IN (
   SELECT OpportunityId FROM OpportunityLineItem WHERE totalPrice > 10000
)
```

Because a great deal of processing work is required for semi-join and anti-join queries, salesforce.com imposes the following restrictions to maintain the best possible performance:

Basic limit:

- No more than two IN or NOT IN statements per WHERE clause.
- You cannot use the NOT operator as a conjunction with semi-joins and anti-joins. Using them converts a semi-join to an anti-join, and vice versa. Instead of using the NOT operator, write the query in the appropriate semi-join or anti-join form.

• Main query limit: In the main WHERE clause, the left operand of any semi-join or anti-join query must query a single primary ID field. However, the selected field in a subquery can be a foreign key. For example:

```
SELECT Id
FROM Idea
WHERE (Id IN (SELECT ParentId FROM Vote WHERE CreatedDate > LAST WEEK))
```

- Subquery limits:
 - A subquery must query a field referencing the same object type as the main query.
 - There is no limit on the number of records matched in a subquery. Standard SOQL query limits apply to the main query.
 - The selected column in a subquery must be a foreign key field, and cannot traverse relationships. This means that you cannot use dot notation in a selected field of a subquery. For example, the following query is valid:

```
SELECT Id, Name
FROM Account
WHERE Id in (
    SELECT AccountId from Contact WHERE LastName LIKE 'Brown_%'
)
```

Using Account.Id (dot notation) instead of AccountId is not supported. Similarly, subqueries like Contact.AccountId FROM Case are invalid.

- You cannot query on the same object in a subquery as in the main query. You can write such *self semi-join queries* without using semi-joins or anti-joins. For example, the following self semi-join query is invalid:

```
SELECT Id, Name
FROM Account
WHERE Id In (
   SELECT ParentId
   FROM Account
   WHERE Name = 'myaccount'
)
```

However, it is very simple to rewrite the query in a valid form, for example:

```
SELECT Id, Name
FROM Account
WHERE Parent.Name = 'myaccount'
```

- You cannot nest a semi-join or anti-join statement in another semi-join or anti-join statement.
- You can use semi-joins and anti-joins in the main WHERE statement, but not in a subquery WHERE statement. For example, the following query is valid:

```
SELECT Id
FROM Idea
WHERE (Idea.Title LIKE 'Vacation%')
AND (Idea.LastCommentDate > YESTERDAY)
AND (Id IN (SELECT ParentId FROM Vote WHERE CreatedById = '005x000000sMgYAAU'))
```

The following query is invalid since the nested query is an additional level deep:

```
SELECT Id
FROM Idea
WHERE
 ((Idea.Title LIKE 'Vacation%')
AND (CreatedDate > YESTERDAY)
```

AND (Id IN (SELECT ParentId FROM Vote WHERE CreatedById = '005x000000sMgYAAU')
)
OR (Idea.Title like 'ExcellentIdea%'))

- You cannot use subqueries in conjunction with OR.
- COUNT, FOR UPDATE, ORDER BY, and LIMIT are not supported in subqueries..
- The following objects are not currently supported in subqueries:
 - ActivityHistory
 - Attachments
 - Event
 - EventAttendee
 - Note
 - OpenActivity
 - Tags (AccountTag, ContactTag, and all other tag objects)
 - Task

Logical Operators

The following table lists the logical operator values that are used in *fieldExpression* syntax:

Operator	Syntax	Description
AND	fieldExpressionX AND fieldExpressionY	true if both fieldExpressionX and fieldExpressionY are true.
OR	fieldExpressionX OR fieldExpressionY	<pre>true if either fieldExpressionX or fieldExpressionY is true. Relationship queries with foreign key values in an OR clause behave differently depending on the version of the API. In a WHERE clause using OR, if the foreign key value in a record is null, the record is returned in Version 13.0 and later, but not returned in versions before 13.0. SELECT Id FROM Contact WHERE LastName = 'foo' or Account.Name = 'bar' The contact with no parent account has a last name that meets the criteria, so it is returned in version 13.0 and later.</pre>
NOT	not fieldExpressionX	true if <i>fieldExpressionX</i> is false. There is also a comparison operator NOT IN.

Quoted String Escape Sequences

You can use the following escape sequences with SOQL:

Sequence	Meaning
\n	New line
\r	Carriage return
\t	Tab
/b	Bell

Sequence	Meaning
\f	Form feed
\"	One double-quote character
\'	One single-quote character
11	Backslash

If you use a backslash character in any other context, an error occurs.

Reserved Characters

Reserved characters, if specified in a SELECT clause as a literal string (between single quotes), must be escaped (preceded by the backslash \ character) in order to be properly interpreted. An error occurs if you do not precede reserved characters with a backslash.

The following characters are reserved:

' (single quote)
\ (backslash)

For example, to query the Account Name field for "Bob's BBQ," use the following SELECT statement:

SELECT Id FROM Account WHERE Name LIKE 'Bob\'s BBQ'

count()

In order to discover the number of rows that are returned by a query, use count () in a SELECT clause:

SELECT count() FROM objectType [WHERE conditionExpression][LIMIT number_of_rows]

For example:

SELECT count() FROM Account WHERE Name LIKE 'a%'

SELECT count() FROM Contact, Contact.Account WHERE Account.name = 'MyriadPubs'

count () must be alone in the SELECT statement, it cannot be mingled with other elements in the SELECT clause. count () cannot be used with ORDER BY, however, it can be used with LIMIT.

The query result ${\tt Size}$ field returns the number of rows. The records themselves are returned as null.

ORDER BY

You can use ORDER BY in a SELECT statement:

```
SELECT fieldList FROM objectType [WHERE conditionExpression]
[ORDER BY fieldName ASC | DESC ? NULLS FIRST | LAST ? ]
[LIMIT number_of_rows]
```

Syntax	Description
ASC or DESC	Specifies whether the results are ordered in ascending (ASC) or descending (DESC) order. Default order is ascending.
NULLS FIRST or NULLS LAST	Orders null records at the beginning (NULLS FIRST) or end (NULLS LAST) of the results. By default, null values are sorted first.

For example, the following query returns a query result with Account records in alphabetical order by first name, sorted in descending order, with accounts that have null names appearing last:

SELECT Name FROM Account ORDER BY Name DESC NULLS LAST

The following factors affect results returned with ORDER BY:

- Sorting is case insensitive.
- ORDER BY is compatible with Relationship Queries syntax.
- Multiple column sorting is supported, by listing more than one *fieldName* clause.
- Relationship queries with foreign key values in an ORDER BY clause behave differently depending on the version of the Force.com API. In an ORDER BY clause, if the foreign key value in a record is null, the record is returned in Version 13.0 and later, but not returned in versions before 13.0.

SELECT Id, CaseNumber, Account.Id, Account.Name FROM Case ORDER BY Account.Name

Any case record for which AccountId is empty is returned in Version 13.0 and later.

The following limitations apply to data types when using ORDER BY:

- These data types are not supported: reference, multi-select picklist, and long text area.
- All other data types are supported, with the following caveats:
 - convertCurrency() always sorts using corporate currency value, if available.
 - phone data does not include any special formatting when sorting, for example, non-numeric characters such as dash or parentheses are included in the sorting.
 - picklist sorting is defined by the picklist sort determined during setup.

You can use ORDER BY with the optional LIMIT qualifier, in a SELECT statement:

SELECT Name FROM Account WHERE industry = 'media' ORDER BY BillingPostalCode ASC NULLS last LIMIT 125

You are limited to 32 fields in an ORDER BY query. If you exceed the limit, a malformed query fault is returned.

LIMIT

Use LIMIT to specify the maximum number of rows to return:

SELECT fieldList FROM objectType [WHERE conditionExpression]
LIMIT number of rows

For example:

SELECT Name FROM Account WHERE industry = 'media' LIMIT 125

This query returns the first 125 records whose industry is Media.

You can use LIMIT with count () as the *fieldList* to count up to the maximum specified.

Example SELECT Clauses

Type of Search	Example(s)	
Simple query	SELECT Id, Name, BillingCity FROM Account	

Type of Search	Example(s)
WHERE	SELECT Id FROM Contact WHERE Name like 'A%' AND MailingCity='California'
ORDER BY	SELECT Name FROM Account ORDER BY Name DESC NULLS LAST
LIMIT	SELECT Name FROM Account WHERE industry = 'media' LIMIT 125
ORDER BY with LIMIT	SELECT Name FROM Account WHERE industry = 'media' ORDER BY BillingPostalCode ASC NULLS last LIMIT 125
count()	SELECT count() FROM Contact
Relationship queries: child-to-parent	SELECT Contact.Firstname, Contact.Account.Name FROM Contact
	SELECT Id, Name, Account.Name FROM Contact WHERE Account.Industry = 'media'
Relationship queries: parent-to-child	SELECT Name, (SELECT LastName FROM Contacts) FROM Account
	SELECT Account.Name, (SELECT Contact.LastName FROM Account.Contacts) FROM Account
Relationship query with WHERE	SELECT Name, (SELECT lastname FROM Contacts WHERE CreatedBy.Alias = 'x') FROM Account WHERE industry = 'media'
Relationship query: child-to parent with custom objects	SELECT Id, FirstName_c, Mother_of_Child_r.FirstName_c FROM Daughter_c WHERE Mother_of_Child_r.LastName_c LIKE 'C%'
Relationship query: parent to child with custom objects	SELECT Id, Name, (SELECT Id, FirstName, LastName FROM Contacts) FROM Account WHERE Name like 'Acme%'
Relationship queries with polymorphic key	SELECT Id, Owner.Name FROM Task WHERE Owner.FirstName like 'B%'
	SELECT Id, Who.FirstName, Who.LastName FROM Task WHERE Owner.FirstName LIKE 'B%'
	SELECT Id, What.Name FROM Event
Relationship queries with aggregate	SELECT Name, (SELECT CreatedBy.Name FROM Notes) FROM Account
	SELECT Amount, Id, Name, (SELECT Quantity, ListPrice, PricebookEntry.UnitPrice, PricebookEntry.Name FROM OpportunityLineItems) FROM Opportunity

Changing the Batch Size in Queries

By default, the number of rows returned in the query result object (batch size) returned in a Query () or call is set to 500. Client applications can change this setting by specifying the batch size in the SetSOAPHeader () call portion of the SOAP header before invoking the Query () call. The maximum batch size is 2,000 records. However this setting is only a suggestion. There is no guarantee that the requested batch size will be the actual batch size. This is done to maximize performance. Although the Office Toolkit handles batching for you, large batch sizes can result in overly large memory signatures, while small batch sizes can result in an excessive number of API requests. To adjust batch size, use the SetSOAPHeader() call.

toLabel()

A client application can have results from a query returned that are translated into the user's language, using toLabel():

toLabel (object.field)

Use toLabel() on regular, multi-select, division, or currency code picklist fields (any field that has picklist values returned by the relevant describe call), or RecordType names. Any organization can use toLabel(). It is particularly useful for organizations that have Translation Workbench enabled.

For example:

SELECT Company, toLabel (Recordtype.Name) FROM Lead

This query returns lead records with the record type name translated into the language for the user who issued the query.



Note: You cannot filter on the translated name value from a record type. Always filter on the master value or the ID of the object for record types.

You can use toLabel() to filter records using a translated picklist value. For example:

SELECT Company, toLabel(Status) from LEAD WHERE toLabel(Status)='le Draft'

Lead records are returned where the picklist value for Status is 'le Draft.' The comparison is made against the value for the user's language. If no translation is available for the user's language for the specified picklist, the comparison is made against the master values.



Note: The tolabel() method cannot be used with ORDER BY. Salesforce always uses the picklist's defined order, just like reports. Also, you can't use tolabel() in the WHERE clause for division or currency ISO code picklists.

Querying Currency Fields in Multicurrency Organizations

If an organization is multicurrency enabled, you can use convertCurrency () in the SELECT clause to convert currency fields to the user's currency.

Use this syntax for the SELECT clause:

convertCurrency(Object)

For example:

SELECT ID, convertCurrency(AnnualRevenue) FROM Account

If an organization has enabled advanced currency management, dated exchange rates will be used when converting currency fields on opportunities, opportunity line items, and opportunity history.

You cannot use the convertCurrency () function in a WHERE clause. If you do, an error is returned. Use the following syntax to convert a numeric value to the user's currency, from any active currency in your organization:

```
WHERE Object_name Operator ISO_CODEvalue
```

For example:

SELECT ID, Name FROM Opportunity WHERE Amount > USD5000

In this example, opportunity records will be returned if the record's currency Amount value is greater than the equivalent of USD5000. For example, an opportunity with an amount of USD5001 would be returned, but not JPY7000.

Use an ISO code that your organization has enabled and is active. If you do not put in an ISO code, then the numeric value is used instead of comparative amounts. Using the example above, opportunity records with JPY5001, EUR5001, and USD5001 would be returned. Note that if you use IN in a WHERE clause, you cannot mix ISO code and non-ISO code values.



Note: Ordering is always based on the converted currency value, just like in reports. Thus, convertCurrency() cannot be used with ORDER BY.

Date Formats and Date Literals

When you specify a date in a SOQL query, it can be a specific date, or a date literal, which is a fixed expression representing a relative range of time such as last month or next year. Remember that dateTime field values are stored as Coordinated Universal Time (UTC). When one of these values is returned in the Salesforce application, it is automatically adjusted for the timezone specified in your organization preferences. Your application may need to handle this conversion.

Date Formats

A *fieldExpression* uses different date formats for date or dateTime fields. If you specify a dateTime format in a query, you can filter on dateTime fields only. Similarly, if you specify a date format value, you can filter on date fields only:

Format	Format Syntax	Example
Date only	YYYY-MM-DD	1999-01-01
Date, time, and time zone offset	 YYYY-MM-DDThh:mm:ss+hh:mm YYYY-MM-DDThh:mm:ss-hh:mm YYYY-MM-DDThh:mm:ssZ 	 1999-01-01T23:01:01+01:00 1999-01-01T23:01:01-08:00 1999-01-01T23:01:01Z

The zone offset is always from UTC. For more information, see:

- http://www.w3.org/TR/xmlschema-2/#isoformats
- http://www.w3.org/TR/NOTE-datetime



Note: For a *fieldExpression* that uses date formats, the date is not enclosed in single quotes. No quotes should be used around the date. For example:

SELECT Id FROM Account WHERE CreatedDate > 2005-10-08T01:02:03Z

Date Literals

A *fieldExpression* can use a date literal to compare a range of values to the value in a date or dateTime field. Each literal is a range of time beginning with midnight (12:00:00). To find a value within the range, use =. To find values on either side of the range, use > or <. The following table shows the available list of date literals, the ranges they represent, and examples:

Date Literal	Range	Example
YESTERDAY	Starts 12:00:00 the day before and continues for 24 hours.	SELECT Id FROM Account WHERE CreatedDate = YESTERDAY
TODAY	Starts 12:00:00 of the current day and continues for 24 hours.	SELECT Id FROM Account WHERE CreatedDate > TODAY
TOMORROW	Starts 12:00:00 after the current day and continues for 24 hours.	SELECT Id FROM Opportunity WHERE CloseDate = TOMORROW
LAST_WEEK	Starts 12:00:00 on the first day of the week before the most recent first day of the week and continues for seven full days. First day of the week is determined by your locale.	SELECT Id FROM Account WHERE CreatedDate > LAST_WEEK
THIS_WEEK	Starts 12:00:00 on the most recent first day of the week before the current day and continues for seven full days. First day of the week is determined by your locale.	SELECT Id FROM Account WHERE CreatedDate < THIS_WEEK
NEXT_WEEK	Starts 12:00:00 on the most recent first day of the week after the current day and continues for seven full days. First day of the week is determined by your locale.	SELECT Id FROM Opportunity WHERE CloseDate = NEXT_WEEK
LAST_MONTH	Starts 12:00:00 on the first day of the month before the current day and continues for all the days of that month.	SELECT Id FROM Opportunity WHERE CloseDate > LAST_MONTH
THIS_MONTH	Starts 12:00:00 on the first day of the month that the current day is in and continues for all the days of that month.	SELECT Id FROM Account WHERE CreatedDate < THIS_MONTH
NEXT_MONTH	Starts 12:00:00 on the first day of the month after the month that the current day is in and continues for all the days of that month.	SELECT Id FROM Opportunity WHERE CloseDate = NEXT_MONTH
last_90_days	Starts 12:00:00 of the current day and continues for the last 90 days.	SELECT Id FROM Account WHERE CreatedDate = LAST_90_DAYS
NEXT_90_DAYS	Starts 12:00:00 of the current day and continues for the next 90 days.	SELECT Id FROM Opportunity WHERE CloseDate > NEXT_90_DAYS
LAST_N_DAYS:n	For the number n provided, starts 12:00:00 of the current day and continues for the last n days.	SELECT Id FROM Account WHERE CreatedDate = LAST_N_DAYS:365
NEXT_N_DAYS:n	For the number n provided, starts 12:00:00 of the current day and continues for the next n DAYS.	SELECT Id FROM Opportunity WHERE CloseDate > NEXT_N_DAYS:15
THIS_QUARTER	Starts 12:00:00 of the current quarter and continues to the end of the current quarter.	SELECT Id FROM Account WHERE CreatedDate = THIS_QUARTER
last_quarter	Starts 12:00:00 of the previous quarter and continues to the end of that quarter.	SELECT Id FROM Account WHERE CreatedDate > LAST_QUARTER
NEXT_QUARTER	Starts 12:00:00 of the next quarter and continues to the end of that quarter.	SELECT Id FROM Account WHERE CreatedDate < NEXT_QUARTER

Date Literal	Range	Example
NEXT_N_QUARTERS:n	Starts 12:00:00 of the next quarter and continues to the end of the <i>n</i> th quarter.	SELECT Id FROM Account WHERE CreatedDate < NEXT_N_QUARTERS:2
LAST_N_QUARTERS:n	Starts 12:00:00 of the previous quarter and continues to the end of the previous <i>n</i> th quarter.	SELECT Id FROM Account WHERE CreatedDate > LAST_N_QUARTERS:2
THIS_YEAR	Starts 12:00:00 on January 1 of the current year and continues through the end of December 31 of the current year.	SELECT Id FROM Opportunity WHERE CloseDate = THIS_YEAR
LAST_YEAR	Starts 12:00:00 on January 1 of the previous year and continues through the end of December 31 of that year.	SELECT Id FROM Opportunity WHERE CloseDate > LAST_YEAR
NEXT_YEAR	Starts 12:00:00 on January 1 of the following year and continues through the end of December 31 of that year.	SELECT Id FROM Opportunity WHERE CloseDate < NEXT_YEAR
NEXT_N_YEARS:n	Starts 12:00:00 on January 1 of the following year and continues through the end of December 31 of the <i>n</i> th year.	SELECT Id FROM Opportunity WHERE CloseDate < NEXT_N_YEARS:5
LAST_N_YEARS:n	Starts 12:00:00 on January 1 of the previous year and continues through the end of December 31 of the previous <i>n</i> th year.	SELECT Id FROM Opportunity WHERE CloseDate > LAST_N_YEARS:5
THIS_FISCAL_QUARTER	Starts 12:00:00 on the first day of the current fiscal quarter and continues through the end of the last day of the fiscal quarter. The fiscal year is defined in the company profile at Setup ➤ Company Profile ➤ Fiscal Year.	SELECT Id FROM Account WHERE CreatedDate = THIS_FISCAL_QUARTER
LAST_FISCAL_QUARTER	Starts 12:00:00 on the first day of the last fiscal quarter and continues through the end of the last day of that fiscal quarter. The fiscal year is defined in the company profile at Setup ➤ Company Profile ➤ Fiscal Year.	SELECT Id FROM Account WHERE CreatedDate > LAST_FISCAL_QUARTER
NEXT_FISCAL_QUARTER	Starts 12:00:00 on the first day of the next fiscal quarter and continues through the end of the last day of that fiscal quarter. The fiscal year is defined in the company profile at Setup ➤ Company Profile ➤ Fiscal Year.	SELECT Id FROM Account WHERE CreatedDate < NEXT_FISCAL_QUARTER
NEXT_N_FISCAL_ QUARTERS:n	Starts 12:00:00 on the first day of the next fiscal quarter and continues through the end of the last day of the <i>n</i> th fiscal quarter. The fiscal year is defined in the company profile at Setup ➤ Company Profile ➤ Fiscal Year.	SELECT Id FROM Account WHERE CreatedDate < NEXT_N_FISCAL_QUARTERS:6
LAST_N_FISCAL_ QUARTERS:n	Starts 12:00:00 on the first day of the last fiscal quarter and continues through the end of the last day of the previous <i>n</i> th fiscal quarter. The fiscal year is defined in the company profile at Setup > Company Profile > Fiscal Year .	SELECT Id FROM Account WHERE CreatedDate > LAST_N_FISCAL_QUARTERS:6

Date Literal	Range	Example
THIS_FISCAL_YEAR	Starts 12:00:00 on the first day of the current fiscal year and continues through the end of the last day of the fiscal year. The fiscal year is defined in the company profile at Setup ➤ Company Profile ➤ Fiscal Year.	SELECT Id FROM Opportunity WHERE CloseDate = THIS_FISCAL_YEAR
LAST_FISCAL_YEAR	Starts 12:00:00 on the first day of the last fiscal year and continues through the end of the last day of that fiscal year. The fiscal year is defined in the company profile at Setup ➤ Company Profile ➤ Fiscal Year.	SELECT Id FROM Opportunity WHERE CloseDate > LAST_FISCAL_YEAR
NEXT_FISCAL_YEAR	Starts 12:00:00 on the first day of the next fiscal year and continues through the end of the last day of that fiscal year. The fiscal year is defined in the company profile at Setup ➤ Company Profile ➤ Fiscal Year.	SELECT Id FROM Opportunity WHERE CloseDate < NEXT_FISCAL_YEAR
NEXT_N_FISCAL_ YEARS:n	Starts 12:00:00 on the first day of the next fiscal year and continues through the end of the last day of the <i>n</i> th fiscal year. The fiscal year is defined in the company profile at Setup > Company Profile > Fiscal Year.	SELECT Id FROM Opportunity WHERE CloseDate < NEXT_N_FISCAL_YEARS:3
LAST_N_FISCAL_ YEARS:n	Starts 12:00:00 on the first day of the last fiscal year and continues through the end of the last day of the previous <i>n</i> th fiscal year. The fiscal year is defined in the company profile at Setup \rightarrow Company Profile \rightarrow Fiscal Year .	SELECT Id FROM Opportunity WHERE CloseDate > LAST_N_FISCAL_YEARS:3



Note: If you have defined **Custom Fiscal Years** in the Salesforce user interface, and in any of the FISCAL date literals you specify a range that is outside the years you've defined, an invalid date error is returned.

Minimum and Maximum Dates

Only dates within a certain range are valid. The earliest valid date is 1700-01-01T00:00:00Z GMT, or just after midnight on January 1, 1700. The latest valid date is 4000-12-31T00:00:00Z GMT, or just after midnight on December 31, 4000.



Note: These values are offset by your timezone. For example, in the Pacific timezone, the earliest valid date is 1699-12-31T16:00:00, or 4:00 PM on December 31, 1699.

null

Use the value null to represent null values in SOQL queries.

For example, the following statement would return the account IDs of all events with a non-null activity date:

SELECT AccountId FROM Event WHERE ActivityDate !=null

Filtering on Boolean Fields

To filter on a Boolean field, use the following syntax:

```
where BooleanField = TRUE
where BooleanField = FALSE
```

Relationship Queries

Client applications need to be able to query for more than a single type of object at a time. SOQL provides syntax to support these types of queries, called *relationship queries*, against both standard objects and custom objects.

Relationship queries traverse parent-to-child and child-to-parent relationships between objects to filter and return results. They are similar to SQL joins. You cannot perform arbitrary SQL joins, however. The relationship queries in SOQL must traverse a valid relationship path as defined in the rest of this section.

You can use relationship queries to search for objects of one type based on criteria that applies to objects of another type, for example, "return all accounts created by Bob Jones and the contacts associated with those accounts." There must be a parent-to-child or child-to-parent relationship connecting the objects. You can't write arbitrary queries such as "return all accounts and users created by Bob Jones."

Use the following topics to understand and use relationship queries in SOQL:

- Understanding Relationship Names
- Using Relationship Queries
- Understanding Relationship Names and Custom Objects and Custom Fields
- Understanding Query Results
- Lookup Relationships and Outer Joins
- Identifying Parent and Child Relationships
- Understanding Polymorphic Keys and Relationships
- Understanding Relationship Query Limitations
- Using Relationship Queries with History Objects
- Using Relationship Queries with the Partner WSDL

Understanding Relationship Names

Parent-to-child and child-to-parent relationships exist between many types of objects, for example, Account is a parent of Contact.



To be able to traverse these relationships for standard objects, a relationship name is given to each relationship. The form of the name is different depending on the direction of the relationship:

• For child-to-parent relationships, the relationship name to the parent is the name of the foreign key, and there is a relationshipName property that holds the reference to the parent object. For example, the Contact child object has a child-to-parent relationship to the Account object, so the value of relationshipName in Contact is Account. These relationships are traversed by specifying the parent using dot notation in the query, for example:

SELECT Contact.FirstName, Contact.Account.Name from Contact

This query returns the first names of all the contacts in the organization, and for each contact, the account name associated with (parent of) that contact.

• For parent-to-child relationships, the parent object has a name for the child relationship that is unique to the parent, the plural of the child object name. For example, Account has child relationships to Assets, Cases, and Contacts among other objects, and has a relationshipName for each, Assets, Cases, and Contacts. These relationships can be traversed only in the SELECT clause, using a nested SOQL query. For example:

```
SELECT Account.Name, (SELECT Contact.FirstName, Contact.LastName FROM Account.Contacts) FROM Account
```

This query returns all accounts, and for each account, the first and last name of each contact associated with (the child of) that account.



Caution: You must use the correct naming convention and SELECT syntax for the direction of the relationship. For information about how to discover relationship names via your organization's WSDL, see Identifying Parent and Child Relationships. There are limitations on relationship queries depending on the direction of the relationship. See Understanding Relationship Query Limitations for more information.

Relationship names are somewhat different for custom objects, though the SELECT syntax is the same. See Understanding Relationship Names and Custom Objects and Custom Fields for more information.

Using Relationship Queries

You can query the following relationships using SOQL:

• Query child-to-parent relationships, which are often many-to-one. Specify these relationships directly in the SELECT, FROM, or WHERE clauses using the dot (.) operator.

For example:

SELECT Id, Name, Account.Name FROM Contact WHERE Account.Industry = 'media'

This query returns the ID and name for only the contacts whose related account industry is media, and for each contact returned, the account name.

• Query parent-to-child, which are almost always one-to-many. Specify these relationships using a subquery (enclosed in parentheses), where the initial member of the FROM clause in the subquery is related to the initial member of the outer query FROM clause. Note that in subqueries, you should specify the plural name of the object, as that is the name of the relationship for each object.

For example:

SELECT Name, (SELECT LastName FROM Contacts) FROM Account

The query returns the name for all the accounts, and for each account, the last name of each contact.

• Traverse the parent-to-child relationship as a foreign key in an aggregate query:

For example:

SELECT Name, (SELECT CreatedBy.Name FROM Notes) FROM Account

This query returns the accounts in an organization, and for each account, the name of the account, the notes for those accounts (which can be an empty result set if there were no notes on any accounts) with the name of the user who created each note (if the result set is not empty).

• In a similar example, traverse the parent-to-child relationship in an aggregate query:

```
SELECT Amount, Id, Name, (SELECT Quantity, ListPrice,
PricebookEntry.UnitPrice, PricebookEntry.Name
FROM OpportunityLineItems) FROM Opportunity
```

Using the same query, you can get the values on Product2 by specifying the product family (which points to the field's data):

```
SELECT Amount, Id, Name, (SELECT Quantity, ListPrice,
PriceBookEntry.UnitPrice, PricebookEntry.Name,
PricebookEntry.product2.Family FROM OpportunityLineItems)
FROM Opportunity
```

• WHERE clauses can be placed on any query (including subqueries), and apply to the root element of the FROM clause of the current query. These clauses can filter on any object in the current scope (reachable from the root element of the query), via the parent relationships.

For example:

```
SELECT Name, (SELECT lastname FROM Contacts WHERE CreatedBy.Alias = 'x')
FROM Account WHERE industry = 'media'
```

This query returns the name for all accounts whose industry is media, and for each account returned, returns the last name of every contact whose created by alias is 'x.'

Understanding Relationship Names and Custom Objects and Custom Fields

Custom objects can participate in relationship queries. Salesforce.com ensures that your custom object names, custom field names, and the relationship names associated with them remain unique, even if a standard object with the same name is available now or in the future. This is important in relationship queries, where the query traverses relationships using the object, field, and relationship names.

This section explains how relationship names for custom objects and custom fields are created and used.

When you create a new custom relationship in the Salesforce user interface, you are asked to specify the plural version of the object name, which you use for relationship queries:

Daughter New Relationship							
Ste	p 3. Enter t	the label and name for lookup field		Step 3 of			
			Previous Next	Cancel			
	Field Label	Mother of Child					
R	Child Ielationship Name	Daughters					
	Field Name	Mother_of_Child					
	Description	Relationship Mother of Child.					
	Required	\checkmark					
			Previous Next	Cancel			

Notice that the Child Relationship Name (parent to child) is the plural form of the child object name, in this case Daughters.

Once the relationship is created, notice that it has an API Name, which is the name of the custom field you created, appended by ______ c (underscore-underscore-c):

Custom Field Mother of Child Help for this Page							
Back to Daughter							
Custom Field Definition Detail Edit Set Field-Level Security View Field Accessibility							
Field Information							
Field Label	Mother of Child	Field Name	Mother_of_Child				
Data Type	Master-Detail	Namespace Prefix					
API Name	Mother_of_Childc						
Description	Relationship Mother of Child.						
Created By	Mysti Berry, 10/31/2006 10:25 AM	Modified By	Mysti Berry, 10/31/2006 10:25 AM				
General Options							
Required	\checkmark						
Master-Detail Options							
Related To	Mother	Child Relationship Name	Daughters				
		Related List Label	Daughters				

When you refer to this field via the API, you must use this special form of the name. This prevents ambiguity in the case where salesforce.com may create a standard object with the same name as your custom field. The same process applies to custom objects—when they are created, they have an API Name, the object named appended by _____, which must be used.

When you use a relationship name in a query, you must use the relationship names without the ______c. Instead, append an _____r (underscore underscore r).

For example:

• When you use a child-to-parent relationship, you can use dot notation:

```
SELECT Id, FirstName_c, Mother_of_Child_r.FirstName_c FROM Daughter_c WHERE Mother of Child r.LastName c LIKE 'C%'
```

This query returns the ID and first name of daughter objects, and the first name of the daughter's mother if the mother's last name begins with 'C.'

· Parent-to-child relationship queries do not use dot notation:

SELECT LastName_c, (SELECT LastName_c FROM Daughters_r) FROM Mother_c

The example above returns the last name of all mothers, and for each mother returned, the last name of the mother's daughters.

Understanding Query Results

Query results are returned as nested objects. The primary or "driving" object of the main SELECT query contains query results of subqueries.

For example, you can construct a query using either parent-to-child or child-to-parent syntax:

• Child-to-parent:

```
SELECT Id, FirstName, LastName, AccountId, Account.Name FROM Contact WHERE Account.Name LIKE 'Acme%'
```

This query returns one query result (assuming there were not too many returned records), with a row for every contact that met the WHERE clause criteria.

Parent-to-child:

```
SELECT Id, Name, (SELECT Id, FirstName, LastName FROM Contacts) FROM Account WHERE Name like 'Acme%'
```

This query returns a set of accounts, and within each account, a query result set of Contact fields containing the contact information from the subquery.

Subquery results are like regular query results in that you may need to use to retrieve all the records if there are many children. For example, if you issue a query on accounts that includes a subquery, your client application must handle results from the subquery as well:

- **1.** Perform the query on Account.
- 2. Iterate over the account with .
- 3. For each account object, retrieve the contacts .
- 4. Iterate over the child contacts, using on each contact's .

The following sample illustrates how to process subquery results:

Lookup Relationships and Outer Joins

Beginning with Version 13.0 of the API, relationship queries return records even if the relevant foreign key field has a null value, as you would expect with an outer join. The change in behavior applies to the following types of relationship queries:

• In an ORDER BY clause, if the foreign key value in a record is null, the record is returned in Version 13.0 and later, but not returned in versions before 13.0. For example:

SELECT Id, CaseNumber, Account.Id, Account.Name FROM Case ORDER BY Account.Name

Any case record for which AccountId is empty is returned in Version 13.0 and later.

The following example uses custom objects:

SELECT ID, Name, Parent r.id, Parent r.name FROM Child c ORDER BY Parent r.name

This query returns the Id and Name values of the Child object and the Id and name of the Parent object referenced in each Child, and orders the results by the parent name. In version 13.0 and later, records are returned even if Parent_r.id or Parent_r.name are null. In earlier versions, such records are not returned by the query.

In a WHERE clause using OR, if the foreign key value in a record is null, the record is returned in Version 13.0 and later, but not returned in versions before 13.0. For example, if your organization has one contact with the value of its LastName field equal to foo and the value of itsAccountId field equal to null, and another contact with a different last name and a parent account named bar, the following query returns only the contact with the last name equal to bar:

SELECT Id FROM Contact WHERE LastName = 'foo' or Account.Name = 'bar'

The contact with no parent account has a last name that meets the criteria, so it is returned in version 13.0 and later.

• In a WHERE clause that checks for a value in a parent field, if the parent does not exist, the record is returned in Version 13.0 and later, but not returned in versions before 13.0.. For example:

SELECT Id FROM Case WHERE Contact.Lastname = null

Case record Id values are returned in version 13.0 and later, but are not returned in versions before 13.0.

Identifying Parent and Child Relationships

You can identify parent-child relationships by viewing the ERD diagrams in Data Model. However, not all parent-child relationships are exposed in SOQL, so to be sure you can query on a parent-child relationship by issuing the appropriate describe call. The results contain parent-child relationship information.

You can also examine the Enterprise WSDL for your organization:

• To find the names of child relationships, look for entries that contain the plural form of a child object and end with type="tns:QueryResult". For example, from Account:

In the example above, the child relationship name Contacts is in the entry for its parent Account.

• For the parent of an object, look for a pair of entries such as AccountId and Account, where the ID field represents the parent object referenced by the ID, and the other represents the contents of the record. The parent entry has a non-primitive type="ens:Account".



٠

Note: Not all relationships are exposed in the API. The most reliable method for identifying relationships is to execute a describeSObject() call. You can use the AJAX Toolkit to quickly execute test calls.

For custom objects, look for a pair of entries with the relationship suffix ___r:

```
<complextType name="Mother c">
<complextcontent>
  <extension base="ens:sObject">
   <sequence>
     <element name="Daughters r" nillable="true" minOccurs="0"</pre>
           type="tns:QueryResult"/>
     <element name="FirstName_c" nillable="true" minOccurs="0"</pre>
          type="xsd:string"/>
     <element name="LastName c" nillable="true" minOccurs="0"</pre>
           type="xsd:string"/>
     . . .
    </sequence>
   </extension>
  </complexContent>
</complextType>
<complextType name="Daughter c">
<complextcontent>
 <extension base="ens:s0bject">
   <sequence>
     <element name="Mother_of_Child__c" nillable="true" minOccurs="0"</pre>
           type="tns:ID"/>
     <element name="Mother of Child r" nillable="true" minOccurs="0"</pre>
           type="xsd:string"/>
     <element name="LastName_c" nillable="true" minOccurs="0"
    type="ens:Mother_c"/>
     . . .
    </sequence>
   </extension>
  </complexContent>
 </complextType>
```
Understanding Polymorphic Keys and Relationships

A polymorphic key is an ID that can refer to more than one type of object as a parent. For example, either a contact or a lead can be the parent of a task. In other words, the WhoId field of a task can contain the ID of either a contact or a lead. If an object can have more than one type of object as a parent, the polymorphic key points to a Name object instead of a single object type.

Executing a DescribeSObjects () call returns the Name object, whose field Type contains a list of the possible object types that can parent the queried object. The field in the indicates that the relationship points to the Name object, needed because the relationship is polymorphic. For example, the value in Whold on Task can be a contact or lead.

In order to traverse relationships where the object type of the parent is not known, you can use these fields to construct a query:

• owner: This field represents the object of a parent who owns the child object, regardless of the parent's object type. For example:

SELECT Id, Owner.Name FROM Task WHERE Owner.FirstName like 'B%'

This example query works for tasks whose owners are either calendars or users.

• who: This field represents the object type of the parent associated with the child:

SELECT Id, Who.FirstName, Who.LastName FROM Task WHERE Owner.FirstName LIKE 'B%'

This example query works for tasks whose owners can be either calendars or users, and whose "who" parent can be either contacts or leads.

If you'd like to know the type of object returned in a query, use who. Type. For example:

SELECT Id, Who.Id, Who.Type FROM Task

Using this example, you could also query all the tasks associated with Contacts:

```
SELECT Id, Who.Id, Who.Type
FROM Task
WHERE Who.Type='Contact'
```

• what: This field represents the object type of a parent that is associated with the child where the object represents something other than a person (that is, not a contact, lead, or user):

SELECT Id, What.Name FROM Event

This example query works for events whose parent may be an account or a solution, or any of another number of object types.

You can also use DescribeSObjects () to obtain information about the parents and children of objects. For more information, see DescribeSObjects () and especially, which, if set to true, indicates the field points to a name.

Understanding Relationship Query Limitations

When designing relationship queries, consider these limitations:

- Relationship queries are not the same as SQL joins. You must have a relationship between objects to create a join in SOQL.
- No more than 20 relationships can be specified in a query.
- In each specified relationship, no more than five levels can be specified in a child-to-parent relationship. For example, Contact.Account.Owner.FirstName (three levels).

- In each specified relationship, only one level of parent-to-child relationship can be specified in a query. For example, if the FROM clause specified Account, the SELECT clause could only specify the Contact or other objects at that level. It could not specify a child object of Contact.
- You can query notes and attachments to get information about them, but you cannot filter on the body of the note or attachment. You cannot filter against the content of textarea fields, blobs, or Scontrols in any object. For example, the following query is valid, and returns all account names and the owner ID for any notes associated with the account:

SELECT Account.Name, (SELECT Note.OwnerId FROM Account.Notes) FROM Account

However, the following query is not valid, because it attempts to evaluate information stored in the body of the note:

SELECT Account.Name, (SELECT Note.Body FROM Account.Notes WHERE Note.Body LIKE 'D%') FROM Account

If you remove the WHERE clause, the query is valid and returns the contents of the body of the note:

SELECT Account.Name, (SELECT Note.Body FROM Account.Notes) FROM Account

Using Relationship Queries with History Objects

Custom objects and some standard objects have an associated history object that tracks changes to an object record. You can use relationship queries to traverse a history object to its parent object. For example, the following query returns every history row for Foo_c and displays the name and custom fields of Foo:

```
SELECT OldValue, NewValue, Parent.Id, Parent.name, Parent.customfield_c FROM foo history
```

This example query returns every Foo object row together with the corresponding history rows in nested subqueries:

```
SELECT Name, customfield_c, (SELECT OldValue, NewValue FROM foo_history)
FROM foo_c
```

Using Relationship Queries with the Partner WSDL

The partner WSDL does not contain the detailed type information available in the enterprise WSDL to get the information you need for a relationship query. You must first execute a DescribeSObjects () call, and from the results, gather the information you need to create your relationship query:

- The value for one-to-many relationships, for example, in an Account object, the relationship name for the asset child is Assets.
- The reference fields available for the relevant object, for example, whold, whatld, or ownerld on a Lead, Case, or custom object.

For an example of using the partner WSDL with relationship queries, see examples on developer.force.com (requires login).

Querying Multi-Select Picklists

Client applications use a specific syntax for querying multi-select picklists (in which multiple items can be selected).

Supported Operators

The following operators are supported for querying multi-select picklists:

Operator	Description
=	Equals the specified string.

Operator	Description
!=	Does not equal the specified string.
includes	Includes (contains) the specified string.
excludes	Excludes (does not contain) the specified string.

Semicolon Character

A semicolon is used as a special character to specify AND. For example, the following notation means 'AAA' and 'BBB':

'AAA;BBB'

Specifying AND is used for multi-select picklists when two or more items must be selected.

Examples

In the following example SOQL notation, the query filters on values in the MSP1___c field that are equal to AAA and BBB selected (exact match):

MSP1 c = 'AAA; BBB'

In the following example SOQL notation:

MSP1__c includes ('AAA; BBB', 'CCC')

the query filters on values in the MSP1_c field that contains either of these values:

- AAA and BBB selected.
- CCC selected.

A match will result on any field value that contains 'AAA' and 'BBB' or any field that contains 'CCC'. For example, the following will be matched:

• matches with 'AAA; BBB':

'AAA;BBB' 'AAA;BBB;DDD'

• matches with ' CCC':

'CCC' 'CCC;EEE' 'AAA;CCC'

Syndication Feed SOQL and Mapping Syntax

Syndication feed services use a SOQL query and mapping specification that allows applications to point to sets of objects and individual objects, as well as to traverse relationships between objects. Several options can also be added as query string parameters to filter and control how the data is presented. Syndication feeds can be defined for public sites.

For full information about the limitations on SOQL in query feed definitions, see the Salesforce online help for syndication feeds.

Refresh()

Refreshes object instances in the client application with corresponding data from the database.

Syntax

```
Refresh(SObject4[], async As Boolean);
```

Usage

Client applications call Refresh() to make sure that the local object instances contain the most recent data from the system. The Refresh() call performs an implicit Retrieve() call for every SObject4 instance in the array.

Sample Code—VBA

```
Public Function CreateAccount() As SObject4
On Error GoTo handleError
'create using batch call
   Dim so(0) As SObject4
   Set so(0) = g_sfApi.CreateObject("account")
   so(0)("name") = "Test Array account"
'call batch method
   g_sfApi.Create so, False
'call batch refresh
   g_sfApi.Refresh so, False
   Exit Function
handleError:
   MsgBox g_sfApi.ErrorMessage
End Function
```

Arguments

Name	Туре	Description
SObject4	SObject4[]	Array of SObject4 instances to refresh.
async	boolean	Specifies whether this invocation should be processed asynchronously (true) or not (false).

Event

RefreshFinished

Retrieve()

Retrieves one or more objects based on the specified object IDs.

Syntax

```
QueryResultSet4[] = Retrieve(fieldList As String, EntityType As String, idList, async As
Boolean);
```

Usage

Use the Retrieve () call to retrieve individual records from an object. The client application passes the list of fields to retrieve, the object, and an array of object IDs to retrieve. The Retrieve () call does not return records that have been deleted.

In general, you use Retrieve () when you know in advance the IDs of the records to retrieve. Use Query () instead to obtain records when you do not know the IDs or when you want to specify other selection criteria.

Client applications can use Retrieve() to perform a client-side join. For example, a client application can run a Query() to obtain a set of Opportunity records, iterate through the returned opportunity records, obtain the accountId for each opportunity, and then call Retrieve() to obtain Account information for those accountIds.

Certain objects cannot be retrieved via the API. To retrieve an object via the Retrieve () call, its object must be configured as retrieveable (Retrieveable is true in the SObject4).

Your client application must be logged in with sufficient access rights to retrieve individual objects within the specified object and to retrieve the fields in the specified field list. For more information, see Factors that Affect Data Access.

Sample Code—VBA: Retrieve—Synchronous Example

```
'Sync retrieve of sobjects
'The result is a queryresult because not all ids are guaranteed to be returned.
' For example, some of the query ids may have been deleted,
' or the user may not have access.
Public Function SyncRetrieve(ids() As String)
On Error GoTo handleError
Dim qr As QueryResultSet4
Dim v As Variant
Dim s As SObject4
Set qr = g_sfApi.Retrieve("*", "account", ids, False)
If g_sfApi.Error <> NO_SF_ERROR Then
'Somebody has failed
Debug.Print g_sfApi.ErrorMessage
End If
For Each v In qr
```

```
'loop through the results
'cast to an SObject4 to see more helpful debug info
Set s = v
If s.Error <> NO_SF_ERROR Then
Debug.Print s.ErrorMessage
Else
'use the object
'NOTE you cannot call async methods until this method returns
s.Update
End If
Next v
Exit Function
handleError:
MsgBox g_sfApi.ErrorMessage
End Function
```

Sample Code—VBA: Retrieve—Asynchronous Example

```
Public Function ASyncRetrieve(ids() As String)
On Error GoTo handleError
   g sfApi.Retrieve "*", "account", ids, True
   Exit Function
handleError:
   MsgBox g_sfApi.ErrorMessage
End Function
'Retrieve callback function
Public Sub g_sfApi_RetrieveFinished(qr As QueryResultSet4)
   Dim v As Variant
   Dim s As SObject4
    If g sfApi.Error <> NO SF ERROR Then
        Debug.Print g_sfApi.ErrorMessage
   End If
   For Each v In qr
'loop through the results
'cast to an sobject4 to see more helpful debug info
        Set s = v
        If s.Error <> NO SF ERROR Then
            Debug.Print s.ErrorMessage
```

```
Else

'use the object

'NOTE you cannot call async methods until this method returns

s("name") = "Hello"

's.Update

End If

Next v

End Sub
```

Arguments

Name	Туре	Description
fieldList	string	List of one or more fields in the specified object, separated by commas. You must specify valid field names and must have read-level permissions to each specified field. The fieldList defines the ordering of fields in the QueryResultSet4 array. You can also specify the wildcard (*) to return all fields for the object. Because using the wildcard (*) might return a very large result set that could slow client application performance, use it only when necessary.
EntityType	string	Object from which to retrieve data. The specified value must be a valid object for your organization. For a complete list of objects, see Standard Objects.
idList	ID[]	Array of one or more IDs of the objects to retrieve. You can pass a maximum of 2000 object IDs to the Retrieve () call. For information on IDs, see ID Field Type.
async	boolean	Specifies whether this invocation should be processed asynchronously (true) or not (false).

Event

RetrieveFinished

Search()

Executes a text search in your organization's data.

Syntax

```
QueryResultSet4[] = Search(searchExpr As String, async
As Boolean);
```

Usage

Use Search() to search for objects based on a search string. The Search() call supports searching custom object. For an extensive discussion about the syntax and rules used for text searches, see Salesforce Object Search Language (SOSL).

Certain objects cannot be searched via the API, such as Attachment objects. To search an object via the Search () call, its object must be configured as searchable (sSearchable is true in the SObject4).

Sample Code—VBA

```
Function SyncSearch()
On Error GoTo handleError
    Dim gr As QueryResultSet4
    Dim v As Variant
   Dim s As SObject4
'Run call in sync mode, but consider running in async mode instead
   Set qr = g sfApi.Search("Find {Sample} IN ALL FIELDS", False)
   For Each v In qr
'loop through the results
'cast to an Sobject4 to see more helpful debug info
       Set s = v
'use the object.
'note that different object types might be returned in the result set.
        Debug.Print s.ObjectType
   Next v
   Exit Function
handleError:
   MsgBox Err.Description
End Function
```

Arguments

Name	Туре	Description
StringsearchExpr	string	Search string that specifies the text expression to search for, the scope of fields to search, the list of objects and fields to retrieve, and the maximum number of objects to return. For more information, see Salesforce Object Search Language (SOSL).
async	boolean	$Specifies whether this invocation should be processed asynchronously (\tt true) or not(\tt false).$

Event

SearchFinished

Salesforce Object Search Language (SOSL)

Use the Salesforce Object Search Language (SOSL) to construct simple but powerful text searches for the Search () call. Unlike SOQL, which can only query one object at a time, SOSL allows you to efficiently search text, email, and phone fields for multiple objects at a time with a single query.

Use the following topics to understand the components of a SOSL FIND statement, including syntax and usage:

- About SOSL
- SOSL Typographical Conventions
- SOSL Syntax

- FIND {SearchQuery}
- IN SearchGroup
- **RETURNING** FieldSpec
- WHERE conditionExpression
- ORDER BY clause
- WITH DivisionFilter
- LIMIT *n*
- toLabel()
- Querying Currency Fields in Multicurrency Organizations
- Example Text Searches
- Text Searches in CJK Languages

About SOSL

SOSL allows you to specify the text expression, the scope of fields to search, the list of objects and fields to retrieve, and conditions for selecting rows in the source objects. You pass the entire SOSL expression in the StringsearchExpr parameter of the Search() call.



Note: If your organization has relationship queries enabled, SOSL supports relationship queries. For more information, see Relationship Queries.

Comparing SOSL and SOQL

Like Salesforce Object Query Language (SOQL), SOSL allows you to programmatically search your organization's Salesforce data for specific information. Unlike SOQL, which can only query one object at a time, you can use a single SOSL query to search all objects—including custom objects—to which you have access. The API executes the search within the specified scope and returns to you only the information that is available to you based on the user permissions under which your application has logged in.

- Use SOQL when you want to use the Query () call to select records for a single object based on criteria that you specify.
- Use SOSL when you want to use the Search () call to find records for one or more objects based on a search string that you specify. The Search () call searches most text fields on an object. For more information on the fields that are searched, see Search Scope.

Designing Efficient Text Searches

When designing text searches, do not be too general, or your search will be slow and return too many results to be useful. SOSL syntax allows you to define the search scope, both in the types of columns to search (IN clause) and the objects to search (RETURNING clause). You can also restrict search results with the LIMIT clause. For examples, see the discussions of IN *SearchGroup*, RETURNING *FieldSpec*, and LIMIT *n*.

Search Scope

The Search () call searches most objects (including custom objects) and text fields to which you have access. It does not search the following objects and fields:

- Any elements such as picklists that are defined as not searchable (Searchable is false in the SObject4).
- Number, date, or checkbox fields. To search for such information, use the Query () call instead.
- Textarea fields, unless you use the ALL FIELDS search group.
- Attachment records associated with certain objects, such as Account, Contact, or Opportunity.



Note: The Search () call does not provide specialized search features such as synonym matching or stop words.

SOSL Typographical Conventions

Topics about SOSL use the following typographical conventions:

Convention	Description
FIND Name IN Account	In an example, Courier font indicates items that you should type as shown. In a syntax statement, Courier font also indicates items that you should type as shown, except for question marks and square brackets.
FIND fieldname IN objectname	In an example or syntax statement, italics represent variables. You supply the actual value.
1	In a syntax statement, the comma inside square brackets indicates that the element containing it may be repeated up to the limits for that element.
[ORDER BY conditionexpression]	In a syntax statement, square brackets surround an element that is optional. You may omit the element, or include one, or if a comma is present, more than one of them.

SOSL Syntax

SOSL uses the following syntax:

```
FIND {SearchQuery} [ toLabel()]
[ IN SearchGroup [ convertCurrency(Amount)] ]
[ RETURNING FieldSpec ]
[ WITH DivisionFilter ]
[ LIMIT n ]
```

where:

Syntax	Description
<pre>FIND {SearchQuery}</pre>	Required. Specifies the text (words or phrases) to search for. The search query must be delimited with curly braces.
toLabel()	Optional. Results from a query are returned translated into the user's language.
IN SearchGroup	 Optional. Scope of fields to search. One of the following values: ALL FIELDS NAME FIELDS EMAIL FIELDS PHONE FIELDS If unspecified, then the default is ALL FIELDS. You can
	specify the list of objects to search in the RETURNING <i>FieldSpec</i> clause.

Syntax	Description
convertCurrency(Amount)	Optional. If an organization is multicurrency enabled, converts currency fields to the user's currency.
RETURNING FieldSpec	Optional. Information to return in the search result. List of one or more objects and, within each object, list of one or more fields, with optional values to filter against. If unspecified, then the search results contain the IDs of all objects found. For information about IDs, see ID Field Type.
WITH DivisionFilter	Optional. If an organization uses divisions, efficiently filters all search results based on values for the Division field.
LIMIT n	Optional. Specifies the maximum number of rows returned in the text query. If unspecified, the default is 200, the largest number of rows that can be returned.



Note: SOSL statements cannot exceed 10,000 characters. For SOSL statements that exceed this maximum length, the API returns an SError of MALFORMED_SEARCH; no result rows are returned.

FIND {SearchQuery}

The required FIND clause allows you to specify the word or phrase to search for. A search query includes:

- The literal text (single word or a phrase) to search for
- Optionally, Wildcards
- · Optionally, logical Operators, including grouping parentheses

Searches are evaluated from left to right and use Unicode (UTF-8) encoding. Text searches are case-insensitive. For example, searching for Customer, customer, or CUSTOMER all return the same results.

Note that special types of text expressions (such as macros, functions, or regular expressions) that are evaluated at run time are not allowed in the FIND clause.



Note: The SearchQuery must be delimited with curly braces. This is needed to unambiguously distinguish the search expression from other clauses in the text search.

Single Words and Phrases

A SearchQuery contains two types of text:

- Single Word—A single word, such as test or hello. Words in the SearchQuery are delimited by spaces, punctuation, and changes from letters to digits (and vice-versa). Words are always case insensitive. In Chinese, Japanese, and Korean (CJK), words are also delimited by pairs of CJK-type characters.
- **Phrase**—A collection of words and spaces surrounded by double quotes such as "john smith". Multiple words can be combined together with logical and grouping Operators to form a more complex query. Certain keywords ("and," "or," and "and not") must be surrounded in double quotes if you want to search for those words, otherwise they are interpreted as the corresponding operator.

Wildcards

You can specify the following wildcard characters to match text patterns in your search:

Wildcard	Description
*	Use an asterisk (*) to match one or more characters at the middle or end of your search term. Do not use the asterisk at the beginning of a search term. If you are searching for a literal asterisk in a word or phrase, then escape the asterisk (precede it with the \ character). For example, a search for john* finds items that start with variations on the term john, such as, johnson or johnny. A search for ma* finds items with mary or marty.
š	Use a question mark (?) to match one character at the middle or end of your search term. Do not use the question mark wildcard at the beginning of a search term. For example, a search for jo?n finds items with the word john or joan.

When using wildcards, consider the following issues:

- The more focused your wildcard search, the faster the search results are returned, and the more likely the results will reflect your intention. For example, to search for all occurrences of the word prospect (or prospects, the plural form), it is more efficient to specify prospect* in the search string than to specify a less restrictive wildcard search (such as prosp*) that could return extraneous matches (such as prosperity).
- Tailor your searches to find all variations of a word. For example, to find property and properties, you would specify propert*.
- Punctuation is indexed. To find * or ? inside a phrase, you must enclose your search string in quotation marks and you must escape the special character. For example, "where are you\?" finds the phrase where are you?. The escape character (\) is required in order for this search to work correctly.

Operators

You can use the following special operators to focus your text search.

Operator	Description
" "	Use quotation marks around search terms to find an exact phrase match. This can be especially useful when searching for text with punctuation. For example, "acme.com" finds items that contain the exact text acme.com. A search for "monday meeting" finds items that contain the exact phrase monday meeting.
AND	Finds items that match all of the search terms. For example, john AND smith finds items with both the word john and the word smith. If an operator is not specified, then this is the default operator. Case-insensitive.
OR	Finds items with at least one of the search terms. For example, john OR smith finds items with either john or smith, or both words. Case-insensitive.
AND NOT	Finds items that do not contain the search term. For example, john AND NOT smith finds items that have the word john but not the word smith. Case-insensitive.
()	 Use parentheses around search terms in conjunction with logical operators to group search terms. For example, you can search for: ("Bob" and "Jones") OR ("Sally" and "Smith")—searches for either Bob Jones or Sally Smith. ("Bob") and ("Jones" OR "Thomas") and Sally Smith—searches for Bob Jones or Bob Thomas and Sally Smith.

Reserved Characters

The following characters are reserved:

Reserved characters, if specified in a text search, must be escaped (preceded by the backslash \ character) in order to be properly interpreted. An error occurs if you do not precede reserved characters with a backslash. This is true even if the SearchQuery is enclosed in double quotes.

For example, to search for the following text:

 $\{1+1\}:2$

insert a backslash before each reserved character:

 $\{1 + 1 \}$:2

Example FIND Clauses

Type of Search	Example(s)
Single term examples	Find {MyProspect}
	<pre>Find {mylogin@salesforce.com}</pre>
	Find {find}
	Find {in}
	Find {returning}
	Find {limit}
Single phrase	Find {John Smith}
Term OR Term	Find {MyProspect OR MyCompany}
Term AND Term	Find {MyProspect AND MyCompany}
Term AND Phrase	<pre>Find {MyProspect AND "John Smith"}</pre>
Term OR Phrase	<pre>Find {MyProspect OR "John Smith"}</pre>
Complex query using AND/OR	Find {MyProspect AND "John Smith" OR MyCompany}
	<pre>Find {MyProspect AND ("John Smith" OR MyCompany) }</pre>
Complex query using AND NOT	Find {MyProspect AND NOT MyCompany}
Wildcard search	<pre>Find {My*}</pre>
Escape sequences	Find {Why not\?}
Invalid or incomplete phrase (will not succeed)	Find {"John Smith}

FIND Clauses in Apex

Note that the syntax of the FIND clause in Apex differs from the syntax of the FIND clause in the Force.com Web Services API:

• In Apex, the value of the FIND clause is demarcated with single quotes. For example:

FIND 'map*' IN ALL FIELDS RETURNING Account (id, name), Contact, Opportunity, Lead

• In the Force.com API, the value of the FIND clause is demarcated with braces. For example:

FIND {map*} IN ALL FIELDS RETURNING Account (id, name), Contact, Opportunity, Lead

For information about using SOSL in Apex, see www.salesforce.com/us/developer/apexcode/index_CSH.htm#apex_data_types.htm#inline_soql_queries.

IN SearchGroup

The optional IN clause allows you to define the types of fields to search. You can specify one of the following values (note that numeric fields are not searchable). If unspecified, the default behavior is to search all text fields in searchable objects.

Valid SearchGroup Settings

Scope	Description
ALL FIELDS	Search all searchable fields. If the IN clause is unspecified, then this is the default setting.
EMAIL FIELDS	Search only email fields.
NAME FIELDS	Search only name fields. In custom objects, fields that are defined as "Name Field" are searched. In standard and custom objects, name fields have the nNameField property set to true in the SObject4(see the array of the parameter of the for more information).
PHONE FIELDS	Search only phone number fields.
SIDEBAR FIELDS	Search for valid records as listed in the Sidebar drop-down list. Unlike search in the application, the asterisk (*) wildcard is not appended to the end of a search string.

While the IN clause is optional, it is recommended that you specify the search scope unless you need to search all fields. For example, if you're searching only for an email address, you should specify IN EMAIL FIELDS in order to design the most efficient search.

Example IN Clauses

Search Type	Example(s)
No search group	Find {MyProspect}
ALL FIELDS	Find {MyProspect} in ALL FIELDS
EMAIL FIELDS	Find {mylogin@mycompany.com} in EMAIL FIELDS
NAME FIELDS	Find {MyProspect} in NAME FIELDS
PHONE FIELDS	Find {MyProspect} in PHONE FIELDS
SIDEBAR FIELDS	Find {MyProspect} in SIDEBAR FIELDS
Invalid search (will not succeed)	Find {MyProspect} in Accounts

RETURNING FieldSpec

The optional RETURNING clause allows you to specify the information that is returned in the text search result. If unspecified, then the default behavior is to return the IDs of all available objects up to the maximum specified in the LIMIT n clause or 200, whichever is smaller.



Note: Solutions, documents, and products must be specified explicitly in a RETURNING clause to be returned in search results. For example:

find {MyProspect} RETURNING Account, Solution, Product2, Document

Use the RETURNING clause to restrict the results data that is returned from the Search () call. For information on IDs, see ID Field Type.

Syntax

In the following syntax statement, square brackets [] represent optional elements that may be omitted. A comma indicates that the indicated segment can appear more than one time.

```
RETURNING ObjectTypeName
[(FieldList [WHERE conditionExpression] [ORDER BY clause] [LIMIT n])]
[, ObjectTypeName [(FieldList) [WHERE conditionExpression] [ORDER BY clause] [LIMIT n])]]
```

RETURNING can contain the following elements:

Name	Description
ObjectTypeName	Object to return. If specified, then the Search() call returns the IDs of all found objects matching the specified object. Must be a valid SObject4 type. You can specify multiple objects, separated by commas. Objects not specified in the RETURNING clause are not returned by the Search() call. For information on IDs, see ID Field Type.
FieldList	Optional list of one or more fields to return for a given object, separated by commas. If you specify one or more fields, then—in addition to the IDs—the fields are also returned for all found objects. You do not need to specify ID fields, as they are always returned. For information on IDs, see ID Field Type.
WHERE conditionExpression	Optional description of how search results for the given object should be filtered, based on individual field values. If unspecified, the search retrieves all the rows in the object that are visible to the user.
	Note that if you want to specify a WHERE clause, you must include a <i>FieldList</i> with at least one specified field. For example,
	RETURNING Account(WHERE name like 'test')
	is not legal syntax, but
	RETURNING Account (Name, Industry WHERE Name like 'test')
	is.
	See conditionExpression for more information.

Name	Description
ORDER BY clause	Optional description of how to order the returned result, including ascending and descending order, and how nulls are ordered. You can supply more than one ORDER BY clause.
	Note that if you want to specify an ORDER BY clause, you must include a <i>FieldList</i> with at least one specified field. For example,
	RETURNING Account (ORDER BY id)
	is not legal syntax, but
	RETURNING Account (Name, Industry ORDER BY Name)
	is.
LIMIT n	Optional clause that sets the maximum number of records returned for the given object. If unspecified, all matching records are returned, up to the limit set for the query as a whole.
	Note that if you want to specify a LIMIT clause, you must include a <i>FieldList</i> with at least one specified field. For example,
	RETURNING Account (LIMIT 10)
	is not legal syntax, but
	RETURNING Account (Name, Industry LIMIT 10)
	is.



Note: The RETURNING clause affects what data is returned, not what data is searched. The IN clause affects what data is searched.

Example RETURNING Clauses

Search Type	Example(s)
No Field Spec	Find {MyProspect}
One sObject, no fields	Find {MyProspect} RETURNING Contact
Multiple SObject4 objects, no fields	Find {MyProspect} RETURNING Contact, Lead
One SObject4, one or more fields	Find {MyProspect} RETURNING Account(Name)
	<pre>Find {MyProspect} RETURNING Contact(FirstName, LastName)</pre>
Custom sObject	Find {MyProspect} RETURNING CustomObject_c
	<pre>Find {MyProspect} RETURNING CustomObject_c(CustomField_c)</pre>
Multiple SObject4 objects, one or more fields, limits	<pre>Find {MyProspect} RETURNING Contact(FirstName, LastName LIMIT 10), Account(Name, Industry)</pre>
Multiple SObject4 objects, mixed number of fields	<pre>Find {MyProspect} RETURNING Contact(FirstName, LastName), Account, Lead(FirstName)</pre>

Search Type	Example(s)
Unsearchable SObject4 objects	Find {MyProspect} RETURNING RecordType
	Find {MyProspect} RETURNING Pricebook
Invalid SObject4 objects	Find {MyProspect} RETURNING FooBar
Invalid sObject SObject4 field	<pre>Find {MyProspect} RETURNING Contact(fooBar)</pre>
Single object limit	<pre>Find {MyProspect} RETURNING Contact(FirstName, LastName LIMIT 10)</pre>
Multiple object limits and a query limit	<pre>Find {MyProspect} RETURNING Contact(FirstName, LastName LIMIT 20), Account(Name, Industry LIMIT 10), Opportunity LIMIT 50</pre>

WHERE conditionExpression

The optional WHERE clause allows you to filter search results for a given object based on individual field values. If unspecified, the search retrieves all the rows in the object that are visible to the user.

condition Expression

The *conditionExpression* of the WHERE clause uses the following syntax:

```
fieldExpression logicalOperator fieldExpression2 ?
```

The condition expressions in SOSL SELECT statements appear in bold in these examples:

- FIND {test} RETURNING Account (id WHERE createddate = THIS_FISCAL_QUARTER)
- FIND {test} RETURNING Account (id WHERE **cf_c includes('AAA')**)
- You can use parentheses to define the order in which *fieldExpressions* are evaluated. For example, the following expression is true if fieldExpression1 is true and either fieldExpression2 or fieldExpression3 are true:

fieldExpression1 AND (fieldExpression2 OR fieldExpression3)

• However, the following expression is true if either fieldExpression3 is true or both fieldExpression1 and fieldExpression2 are true.

(fieldExpression1 AND fieldExpression2)OR fieldExpression3

• Client applications must specify parentheses when nesting operators. However, multiple operators of the same type do not need to be nested.

See *fieldExpression* for the syntax of *fieldExpression*. See Logical Operators for the valid logical operators.

fieldExpression

A fieldExpression uses the following syntax:

```
fieldName comparisonOperator value
```

where:

Syntax	Description
fieldName	The name of a field in the specified object. Use of single or double quotes around the name will result in an error. You must have at least read-level permissions to the field. It can be any field except a long text area field, encrypted data field, or base64-encoded field. It does not need to be a field in the <i>fieldList</i> .
comparisonOperator	See Comparison Operators for a list of valid operators.
value	A value used to compare with the value in <i>fieldName</i> . You must supply a value whose data type matches the field type of the specified field. You must supply a native value—other field names or calculations are not permitted. For date values, use the formatting listed in Date Formats and Date Literals. If quotes are required (for example, they are not for dates and numbers), use single quotes. Double quotes result in an error.

Comparison Operators

The following table lists the *comparisonOperator* values that are used in *fieldExpression* syntax. Note that comparisons on strings are case-insensitive.

Operator	Name	Description
=	Equals	Expression is true if the value in the specified <i>fieldName</i> equals the specified <i>value</i> in the expression. String comparisons using the equals operator are case-insensitive.
! =	Not equals	Expression is true if the value in the specified <i>fieldName</i> does not equal the specified <i>value</i> .
<	Less than	Expression is true if the value in the specified <i>fieldName</i> is less than the specified <i>value</i> .
<=	Less or equal	Expression is true if the value in the specified fieldName is less than, or equals, the specified value.
>	Greater than	Expression is true if the value in the specified <i>fieldName</i> is greater than the specified <i>value</i> .
>=	Greater or equal	Expression is true if the value in the specified <i>fieldName</i> is greater than or equal to the specified <i>value</i> .
LIKE	Like	 Expression is true if the value in the specified fieldName matches the characters of the text string in the specified value. The LIKE operator in SOQL and SOSL is similar to the SAME operator in SQL; it provides a mechanism for matching partial text strings and includes support for wildcards. The % and _ wildcards are supported for the LIKE operator. The % wildcard matches zero or more characters. The _ wildcard matches exactly one character. The text string in the specified value must be enclosed in single quotes. The LIKE operator is supported for string fields only (see string). The LIKE operator in SOQL and SOSL does not currently support escaping of special characters % or

Operator	Name	Description
		• You should not use the backslash character in a search (except to escape a character), as it is reserved.
		For example, the following query matches Appleton, Apple, and Bappl , but not Appl:
		SELECT AccountId, FirstName, lastname FROM Contact WHERE lastname LIKE 'appl_%'
IN	IN	If the value equals any one of the specified values in a ${\tt WHERE}$ clause. For example:
		SELECT Name FROM ACCOUNT WHERE BillingState IN ('California', 'New York')
		Note that the values for IN must be in parentheses. String values must be surrounded by single quotes.
		IN and NOT IN can also be used for semi-joins and anti-joins when searching on ID fields. For more information, see Semi-Joins with IN and Anti-Joins with NOT IN.
NOT IN	NOT IN	If the value does not equal any of the specified values in a WHERE clause. For example:
		SELECT Name FROM ACCOUNT WHERE BillingState NOT IN ('California', 'New York')
		Note that the values for NOT IN must be in parentheses, and string values must be surrounded by single quotes.
		There is also a logical operator NOT.
INCLUDES EXCLUDES		Applies only to multi-select picklists. See Querying Multi-Select Picklists.

Logical Operators

The following table lists the logical operator values that are used in *fieldExpression* syntax:

Operator	Syntax	Description
AND	fieldExpressionX AND fieldExpressionY	true if both fieldExpressionX and fieldExpressionY are true.
OR	fieldExpressionX OR fieldExpressionY	<pre>true if either fieldExpressionX or fieldExpressionY is true. Relationship queries with foreign key values in an OR clause behave differently depending on the version of the API. In a WHERE clause using OR, if the foreign key value in a record is null, the record is returned in Version 13.0 and later, but not returned in versions before 13.0. SELECT Id FROM Contact WHERE LastName = 'foo' or Account.Name = 'bar'</pre>

Operator	Syntax	Description
		The contact with no parent account has a last name that meets the criteria, so it is returned in version 13.0 and later.
NOT	not fieldExpressionX	true if fieldExpressionX is false.
		There is also a comparison operator NOT IN.

Quoted String Escape Sequences

You can use the following escape sequences with SOSL:

Sequence	Meaning
\n	New line
\r	Carriage return
\t	Tab
\b	Bell
\f	Form feed
\ "	One double-quote character
\ '	One single-quote character
	Backslash

If you use a backslash character in any other context, an error occurs.

Example WHERE Clauses

Example(s)
FIND {test} RETURNING Account (id WHERE createddate = THIS_FISCAL_QUARTER)
FIND {test} RETURNING Account (id WHERE cfc includes('AAA'))
<pre>FIND {test} RETURNING Account (id), User(Field1,Field2 WHERE Filed1 = 'test' order by id ASC, Name DESC)</pre>
<pre>FIND {test} IN ALL FIELDS RETURNING Contact(Salutation, FirstName, LastName, AccountId WHERE Name = 'test'), User(FirstName, LastName), Account(id WHERE BillingState IN ('California', 'New York'))</pre>
<pre>FIND {test} RETURNING Account (id WHERE (FirstName = 'New Account' and ((not AccountId = null) or AccountId != null)) or (AccountId = '001z0000008Vq7' and FirstName = 'Account Insert Test') or (NumberOfEmployees < 100 or NumberOfEmployees = null) ORDER BY NumberOfEmployees)</pre>

ORDER BY clause

You can use one or more ORDER BY clauses in a SOSL statement.

Syntax

ORDER BY fieldname [ASC | DESC] [NULLS [first | last]

Syntax	Description
ASC or DESC	Ascending (ASC) or descending (DESC) order of the results. Default order is ascending.
NULLS	Order null records at the beginning (first) or end (last) of the results. By default, NULL values are sorted last in ascending order and first in descending order.

Examples

The following example shows a single ORDER BY clause:

FIND {MyName} RETURNING Account (Name, Id ORDER BY Id)

The following example shows multiple ORDER BY clauses:

```
FIND {MyContactName} RETURNING Contact(Name, Id ORDER BY Name), Account(Description, Id
ORDER BY Description)
```

The following search returns a result with account records in alphabetical order by name, sorted in descending order, with accounts that have null names appearing last:

```
FIND {MyAccountName} IN NAME FIELDS RETURNING Account(Name, Id ORDER BY Name DESC NULLS
last)
```

WITH DivisionFilter

The optional WITH clause allows you to filter all search results based on division. Although you can also filter on an object's Division field within a WHERE clause, using WITH is preferable because:

- It pre-filters all records based on division before applying other filters
- You can specify the division's name in the filter, rather than its ID (as is required if you filter on division in the WHERE clause)

For example:



Note: Users can perform searches based on division regardless of whether they have the "Affected by Divisions" permission enabled.

LIMIT n

The optional LIMIT clause allows you to specify the maximum number of rows returned in the text query. If unspecified, then the default is 200, which is the largest number of rows that can be returned.

You can set limits on individual objects, or on an entire query. Setting individual object limits allows you to prevent results from a single object using up the maximum query limit before other objects are returned. For example, if you issue the following query, at most 20 account records can be returned, and the remaining number of records can be contacts.

FIND {test} RETURNING Account(id LIMIT 20), Contact LIMIT 100

Object limits are evaluated in the order they are included in the query, and the maximum query limit is adjusted after each object is processed. For example, if only seven accounts match the query string above, then at most 93 contact records can be returned. Likewise, if the following query returns 15 accounts and 30 contacts, then only 55 opportunities can be returned, regardless of the Opportunity object's limit of 75:

FIND {test} RETURNING Account(id LIMIT 20), Contact, Opportunity(id LIMIT 75) LIMIT 100

If you specify a limit of 0, no records are returned for that object.

toLabel()

A client application can have results from a query returned that are translated into the user's language, using tolabel():

```
toLabel(object.field)
```

For example:

FIND {Joe} RETURNING Lead(company, toLabel(Recordtype.Name))

This query returns lead records with the record type name translated into the language for the user who issued the query.



Note: You cannot filter on the translated name value from a record type. Always filter on the master value or the ID of the object for record types.

You can use toLabel () to filter records using a translated picklist value. For example:

```
FIND {test} RETURNING Lead(company, toLabel(Status) WHERE toLabel(Status)
= 'le Draft' )
```

Lead records are returned where the picklist value for Status is 'le Draft.' The comparison is made against the value for the user's language. If no translation is available for the user's language for the specified picklist, the comparison is made against the master values.



Note: The toLabel() method cannot be used with the ORDER BY *clause*. Salesforce always uses the order defined in the picklist, just like reports.

Querying Currency Fields in Multicurrency Organizations

If an organization is multicurrency enabled, you can use convertCurrency() in the *FieldList* of the RETURNING clause to convert currency fields to the user's currency.

convertCurrency(Amount)

For example,

FIND {test} RETURNING Opportunity(Name, convertCurrency(Amount))

If an organization has enabled advanced currency management, dated exchange rates will be used when converting currency fields on opportunities, opportunity line items, and opportunity history.

You cannot use the convertCurrency () function in a WHERE clause. If you do, an error is returned. Use the following syntax to convert a numeric value to the user's currency, from any active currency in your organization:

WHERE Object name Operator ISO CODEvalue

For example:

FIND {test} IN ALL FIELDS RETURNING Opportunity (Name WHERE Amount>USD5000)

In this example, opportunity records will be returned if the record's currency Amount value is greater than the equivalent of USD5000. For example, an opportunity with an amount of USD5001 would be returned, but not JPY7000.

Use an ISO code that your organization has enabled and is active. If you do not put in an ISO code, then the numeric value is used instead of comparative amounts. Using the example above, opportunity records with JPY5001, EUR5001, and USD5001 would be returned. Note that if you use IN in a WHERE clause, you cannot mix ISO code and non-ISO code values.



Note: Ordering is always based on the converted currency value, just like in reports. Thus, convertCurrency() cannot be used with the ORDER BY *clause*.

Example Text Searches

Look for joe anywhere in the system. Return the IDs of the records where joe is found.

Find {joe}

Look for the name Joe Smith anywhere in the system, in a case-insensitive search. Return the IDs of the records where Joe Smith is found.

Find {Joe Smith}

Look for the name JOE Smith in the name field of a lead, return the ID field of the records.

Find {Joe Smith} In Name Fields Returning lead

Look for the name Joe Smith in the name field of a lead and return the name and phone number.

```
Find {Joe Smith}
In Name Fields
Returning lead(name, phone)
```

Look for the name Joe Smith in the name field of a lead and return the name and phone number of any matching record that was also created in the current fiscal quarter.

```
Find {Joe Smith}
In Name Fields
Returning lead (name, phone Where createddate = THIS FISCAL QUARTER)
```

Look for the name Joe Smith or Joe Smythe in the name field of a lead or contact and return the name and phone number. If an opportunity is called Joe Smith, the opportunity should not be returned.

```
Find {"Joe Smith" OR "Joe Smythe"}
In Name Fields
Returning lead(name, phone), contact(name, phone)
```

Wildcards:

Find {Joe Sm*}
Find {Joe Sm?th*}

Delimiting "and" and "or" as literals when used alone:

```
Find {"and" or "or"}
Find {"joe and mary"}
Find {in}
Find {returning}
Find {find}
```

Escaping special characters & $|!() \{\} []^{*} ~ * ? : \$

```
Find {right brace \}}
Find {asterisk \*}
Find {question \?}
Find {single quote \'}
Find {double quote \"}
```

Text Searches in CJK Languages

In Chinese, Japanese, and Korean (CJK), words are delimited by pairs of CJK-type characters.

SetProxyInfo()

Sets the proxy settings programmatically.

Syntax

```
SetProxyInfo(username, password, url, configUrl);
```

Usage

If the client application needs to set the proxy settings programmatically instead of letting Office Toolkit handle them automatically.

Sample Code—VBA

```
'Only API version 7.0 is supported for this release
Function SetProxyInfo(username As String, password As String, url As String,
configUrl as String)
g_sfApi.SetProxyInfo "myname", "mypassword",
         "http://myorg.com.proxy_file.txt",
         "http://myorg.com.config_URL"
End Function
```

Arguments

Name	Туре	Description
username	string	Username for the login.
password	string	Password for username login.
url	string	Direct proxy URL.
configUrl	string	Proxy script URL.

When a script URL is set but the proxy address cannot be accessed, for example, the address is only available inside a corporate network but the user is logging in from home, Office Toolkit will use the direct URL if it has been set, or try a direct connection if the direct URL has not been set.

If a direct URL is set and it cannot be accessed, Office Toolkit will not try a direct connection. This is the same behavior as Internet Explorer.

SetServerURL()

Sets the endpoint URL prior to login.

Syntax

SetServerUrl(url As String);

Usage

On rare occasions, client applications might need to set the URL endpoint before calling Login (). For example, a developer may wish to try out a new version of the API before it is generally available.

Sample Code—VBA

```
'Only API version 6.0 is supported for this release
Function SetServerUrl(url As String)
    g_sfApi.SetServerUrl "https://test.salesforce.com/services/Soap/c/"
End Function
```

Arguments

Name	Туре	Description
url	string	URL to use as the endpoint.

SetSOAPHeader()

Sets the SOAP header for configuring SOAP header options, such as the batch size for Query () calls or lead assignment rules.

Syntax

```
SetSoapHeader(section As String, Name
As String, Value As String)
```

Usage

Client applications use this API call to set various options in the SOAP header. Supported options are described below.

Sample Code—VBA

```
Function SetSoapHeader(section As String, key As String, value As String)
```

'Change the query batch size to be the max

g sfApi.SetSoapHeader "QueryOptions", "batchSize", "2000"

End Function

Arguments

Name	Туре	Query Batchsize	Assignment Rule	Default Assignment Rule
Section	string	QueryOptions	AssignmentRuleHeader	AssignmentRuleHeader
Name	string	batchSize	assignmentRuleId	useDefaultRule
Value	string	Specify the number of records to return in a Query () call. Default is 2000.	ID of a specific assignment rule to run for the Case or Lead. Can be an inactive assignment rule. The ID can be retrieved by querying the AssignmentRule object (for details, see AssignmentRule.) If specified, do not specify useDefaultRule.	If true, uses the default (active) assignment rule for a Case or Lead. If specified, do not specify an assignmentRuleId.

Undelete()

Undeletes objects from the recycle bin.

Syntax

```
Undelete(SObject4[], async As Boolean);
```

Usage

Use this call to restore any deleted record that is undeletable. Undeletable records include those in the recycle bin.

You should verify that an object can be undeleted before attempting to delete it. Some objects cannot be undeleted, for example, Account objects can be undeleted, but not AccountTeamMember objects.

Since a delete call cascade-deletes child records, an undelete call will undelete the cascade-deleted records. For example, deleting an account will delete all the contacts associated with that account.

You can undelete records that were deleted as the result of a merge, but the child objects will have been re-parented, which cannot be undone.

Sample Code—VBA

Event

UndeleteFinished

Arguments

Name	Туре	Description
ids	ID[]	IDs of the objects to be restored.

Update()

Updates one or more existing objects in your organization's data.

Syntax

```
Update(SObject4[], async As Boolean);
```

Usage

Use this call to update one or more existing objects, such as individual accounts or contacts, in your organization's data. The Update () call is analogous to the UPDATE statement in SQL.

Permissions

Your client application must be logged in with sufficient access rights to Update () individual objects (as well as individual fields inside that object) within the specified object. For more information, see Factors that Affect Data Access.

Special Handling

Certain objects—and certain fields within those objects—require special handling or permissions. For example, you might also need permissions to access this object's parent object. Before you attempt to update a particular object, be sure to read its description in the Standard Objects and in the Salesforce online help.

Updateable Objects

Certain objects cannot be updated via the API. To update an object via the Update() call, its object must be configured as updateable (Updateable is true in the SObject4).

Required Fields

When updating required fields, you must supply a value—you cannot set the value to null. For more information, see Required Fields.

ID Fields

Fields whose names contain "Id" are either that object's primary key (see ID Field Type) or a foreign key (see Reference Field Type). Client applications cannot update primary keys, but they can update foreign keys. For example, a client application can update the <code>OwnerId</code> of an Account, because <code>OwnerID</code> is a foreign key that refers to the user who owns the account record. View the properties of the Field4 to confirm whether the field can be updated.

Automatically Updated Fields

The API updates certain fields automatically, such as LastModifiedDate, LastModifiedById, and SystemModstamp. You cannot explicitly specify these values in your Update() call.

Resetting Values to null

To reset a field value to null, you set the field value to VT_EMPTY. You cannot set required fields (nillable is false in the SObject4) to null.

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 specified for your programming language and development tool (your development tool will handle 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 name field is entered as "ABC Company ", then the value is stored in the database as "ABC Company".

Assignment Rules

When updating Case or Lead objects, your client application can set SOAP header (see SetSOAPHeader()) options to have the case or lead automatically assigned to one or more users based on assignment rules configured in the Salesforce user interface. For more information, see Case or Lead.

Maximum Number of Objects Created

Your client application can change up to 200 individual objects in a single Update () call. If an update request exceeds 200 objects, then the entire operation fails.

Update()

You can use external ID fields as a foreign key, allowing you to create or update records in a single step instead of querying a record to get the ID first. To do this, specify the foreign key name and the external ID field value. For example:

Basic Steps for Updating Objects

Use this process to update objects:

- 1. Determine the ID of each object that you want to Update(). For example, you might call Query() to retrieve a set of objects (with their IDs), based on specific criteria, that you would want to update. If you know the ID of the object that you want to update, you can call Retrieve() instead. For information on IDs, see ID Field Type.
- 2. For each object, populate its fields with the data that you want to update.
- 3. Construct an SObject4[] array and populate that array with the objects that you want to update. All objects must be of the same object.
- 4. Call Update (), passing in the SObject4[] array.
- 5. Process the results in the SObject4[] property to verify whether the objects have been successfully updated.

Sample Code—VBA: Update—Synchronous Example

```
Function SyncUpdate(account As SObject4)
On Error GoTo handleError
Dim so(0) As SObject4
account("name").Value = "New Value"
'you can call update on the object and through the session
```

```
Core Calls
```

account.Update

'put the account into an array

Set so(0) = account

'call synchronous batch update

g_sfApi.Update so, False

Exit Function

handleError:

MsgBox g_sfApi.ErrorMessage

End Function

```
Function Lookup(entityType As String)
  g_sfApi.DoLookupSearch entityType, ""
End Function
```

Sample Code—VBA: Update—Asynchronous Example

```
Function ASyncUpdate (account As SObject4)
On Error GoTo handleError
   Dim so(0) As SObject4
   account("name").Value = "New Value"
'put the account into an array
  Set so(0) = account
'call asynchronous batch update
   g_sfApi.Update so, True
   Exit Function
handleError:
   MsgBox g_sfApi.ErrorMessage
End Function
'Asynchronous update callback.
'The array contains the same objects as the source update call.
'If an error occured, examine the objects in the array one by one to determine
'which ones failed.
Public Sub g sfApi UpdateFinished(pso As Variant)
    Dim so() As SObject4
    Dim i As Integer
'check session for errors
    If g sfApi.Error <> NO SF ERROR Then
'This does not mean all the sobject in the array failed, only that one or more have failed
```

'You have to check each object to determine if it is in error

```
Debug.Print g_sfApi.ErrorMessage
End If
so = pso
'look at the array
For i = 0 To UBound(so)
If so(i).Error <> NO_SF_ERROR Then
Debug.Print so(i).ErrorMessage
Else
'use the object
'NOTE you cannot call async methods until this method returns
so(i)("name") = "callback mod"
'so(i).Update
End If
Next i
End Sub
```

Arguments

Name	Туре	Description
SObject4	SObject4[]	Array of one or more objects (maximum of 200) to update.
async	boolean	Specifies whether this invocation should be processed asynchronously (true) or not (false).

Event

UpdateFinished

Chapter 10

Describe Calls

The following table lists supported describe calls in the API in alphabetical order, and provides a brief description for each. Click a call name to see syntax, usage, and more information for that call.



Note: For a list of API utility calls, see Utility Calls, and for a list of general calls (calls that query, retrieve, or modify data), see Core Calls.

Call	Description
DescribeGlobal()	Retrieves a list of available objects for your organization's data.
DescribeSObject()	Retrieves metadata (field list and object properties) for the specified object type. Superseded by DescribeSObjects().
DescribeSObjects()	An array-based version of describeSObject.
DescribeTabs()	Describes the apps and tabs that have been configured for the user.

DescribeGlobal()

Retrieves a list of available objects for your organization's data.

Syntax

DescribeGlobalResult = binding.describeGlobal();

Usage

Use DescribeGlobal() to obtain a list of available objects for your organization. You can then iterate through this list and use DescribeSObjects() to obtain metadata about individual objects.

Your client application must be logged in with sufficient access rights to retrieve metadata about your organization's data. For more information, see Factors that Affect Data Access.

Sample Code—Java

```
private void describeGlobalSample() {
  try
  {
    DescribeGlobalResult describeGlobalResult = null;
    describeGlobalResult = binding.describeGlobal();
    String[] types = describeGlobalResult.getTypes();
```

}

```
for (int i=0;i<types.length;i++)
   System.out.println(types[i]);
}
catch (Exception ex)
{
   System.out.println("\nFailed to return types, error message was: \n" +
        ex.getMessage());
}</pre>
```

Sample Code—C#

```
private void globalDescribe()
{
    //Invoke describeGlobal call and save results in DescribeGlobalResult object
    DescribeGlobalResult dgr = binding.describeGlobal();
    // Iterate through the results
    for (int i=0;i<dgr.types.Length;i++)
    {
        // The dgr.types[i] object is a string
        System.Diagnostics.Trace.WriteLine(dgr.types[i]);
    }
    binding.describeSObject
}
</pre>
```

Arguments

None.

Response

DescribeGlobalResult

DescribeGlobalResult

The DescribeGlobal () call returns a DescribeGlobalResult object, which has the following properties.

DescribeSObject()

Describes metadata (field list and object properties) for the specified object.

Syntax

DescribeSObjectResult = binding.describeSObject(string sObjectType);

Usage

Use DescribeSObject() to obtain metadata for a given object. You can first call DescribeGlobal() to retrieve a list of all objects for your organization, then iterate through the list and use DescribeSObject() to obtain metadata about individual objects.

Your client application must be logged in with sufficient access rights to retrieve metadata about your organization's data. For more information, see Factors that Affect Data Access.

Sample Code—Java

```
public void describeSample()
ł
  try {
    // Invoke describeSObject and save results in DescribeSObjectResult
    DescribeSObjectResult describeSObjectResult = binding.describeSObject("Account");
    // Determine whether the describeSObject call succeeded.
    if (! (describeSObjectResult == null)) {
      // Retrieve fields from the results
      Field[] fields = describeSObjectResult.getFields();
      // Get the name of the object
      String objectName = describeSObjectResult.getName();
      // Get some flags
      boolean isActivateable = describeSObjectResult.isActivateable();
      // Many other values are accessible
      if (! (fields == null)) {
        // Iterate through the fields to get properties for each field
        for (int i = 0; i < fields.length; i++) {</pre>
          Field field = fields[i];
          int byteLength = field.getByteLength();
          int digits = field.getDigits();
          String label = field.getLabel();
          int length = field.getLength();
          String name = field.getName();
          PicklistEntry[] picklistValues = field.getPicklistValues();
          int precision = field.getPrecision();
          String[] referenceTos = field.getReferenceTo();
          int scale = field.getScale();
          FieldType fieldType = field.getType();
          boolean fieldIsCreateable = field.isCreateable();
          // Determine whether there are picklist values
          if (picklistValues != null) {
            System.out.println("Picklist values = ");
            for (int j = 0; j < picklistValues.length; j++) {
              if (picklistValues[j].getLabel() != null) {
               System.out.println(" Item: " + picklistValues[j].getLabel());
              }
            }
          }
          // Determine whether this field refers to another object
          if (referenceTos != null) {
            System.out.println("Field references the following objects:");
            for (int j = 0; j < referenceTos.length; j++) {</pre>
             System.out.println(" " + referenceTos[j]);
          }
        }
      }
    }
  } catch (Exception ex) {
    System.out.println("\nFailed to get Account description, error message was: \n" +
                       ex.getMessage());
  }
```

Sample Code—C#

```
private void sObjectDescribe()
{
    //Invoke describeSObject and save results in DescribeSObjectResult
    DescribeSObjectResult dsr = binding.describeSObject("Account");
    //Get value that indicates whether we can create a record
    bool canCreate = dsr.createable;
```

```
//Get a field and save its name
String fldName = dsr.fields[0].name;
```

Arguments

}

Response

DescribeSObjectResult

DescribeSObjectResult

The DescribeSObject() call returns a DescribeSObjectResult object.

The DescribeSObject() call has been deprecated. See DescribeSObjectResult in the describeSObjects section for details.

DescribeSObjects()

An array-based version of DescribeSObject(); describes metadata (field list and object properties) for the specified object or array of objects.



Note: Use this call instead of DescribeSObject().

Syntax

DescribeSObjectResult [] = binding.describeSObjects(string sObjectType[]);

Usage

Use DescribeSObjects () to obtain metadata for a given object or array of objects. You can first call DescribeGlobal () to retrieve a list of all objects for your organization, then iterate through the list and use DescribeSObjects () to obtain metadata about individual objects. The DescribeSObjects () call is limited to a maximum of 100 objects returned.

Your client application must be logged in with sufficient access rights to retrieve metadata about your organization's data. For more information, see Factors that Affect Data Access.

In organizations where person accounts are enabled, this call shows Accounts as not createable if the profile does not have access to any business account record types.

Sample Code—Java

```
private void describeSObjectsSample()
{
  try {
    DescribeSObjectResult[] describeSObjectResults =
    binding.describeSObjects(new String[] {"account", "contact", "lead"});
    for (int x=0;x<describeSObjectResults.length;x++)
    {
        DescribeSObjectResult describeSObjectResult = describeSObjectResults[x];
        // Retrieve fields from the results
        Field[] fields = describeSObjectResult.getFields();
        // Get the name of the object</pre>
```

```
String objectName = describeSObjectResult.getName();
    // Get some flags
   boolean isActivateable = describeSObjectResult.isActivateable();
   System.out.println("Object name: " + objectName);
    // Many other values are accessible
   if (fields != null)
      // Iterate through the fields to get properties for each field
      for (int i = 0; i < fields.length; i++)</pre>
       Field field = fields[i];
       int byteLength = field.getByteLength();
        int digits = field.getDigits();
        String label = field.getLabel();
        int length = field.getLength();
        String name = field.getName();
        PicklistEntry[] picklistValues = field.getPicklistValues();
        int precision = field.getPrecision();
        String[] referenceTos = field.getReferenceTo();
        int scale = field.getScale();
        FieldType fieldType = field.getType();
       boolean fieldIsCreateable = field.isCreateable();
        System.out.println("Field name: " + name);
        // Determine whether there are picklist values
       if (picklistValues != null && picklistValues[0] != null)
          System.out.println("Picklist values = ");
          for (int j = 0; j < picklistValues.length; j++)</pre>
            if (picklistValues[j].getLabel() != null)
            {
              System.out.println(" Item: " +
                  picklistValues[j].getLabel());
            }
          }
        // Determine whether this field refers to another object
        if (referenceTos != null && referenceTos[0] != null)
        {
          System.out.println("Field references the following objects:");
          for (int j = 0; j < referenceTos.length; j++)</pre>
            System.out.println(" " + referenceTos[j]);
        }
     }
   }
 }
} catch (Exception ex) {
 System.out.println("\nFailed to get object descriptions, error message was: \n" +
                     ex.getMessage());
}
```

Sample Code—C#
```
sforce.Field[] fields = describeSObjectResult.fields;
    // Get the name of the object
    String objectName = describeSObjectResult.name;
    // Get some flags
    bool isActivateable = describeSObjectResult.activateable;
    // Many other values are accessible
    if (fields != null)
    {
        // Iterate through the fields to get properties for each field
        for (int i = 0; i < fields.Length; i++)</pre>
        {
            sforce.Field field = fields[i];
            int byteLength = field.byteLength;
            int digits = field.digits;
            string label = field.label;
            int length = field.length;
            string name = field.name;
            sforce.PicklistEntry[] picklistValues = field.picklistValues;
            int precision = field.precision;
            string[] referenceTos = field.referenceTo;
            int scale = field.scale;
            sforce.fieldType fieldType = field.type;
            bool fieldIsCreateable = field.createable;
            // Determine whether there are picklist values
            if (picklistValues != null && picklistValues[0] != null)
            {
                Console.WriteLine("Picklist values = ");
                for (int j = 0; j < picklistValues.Length; j++)</pre>
                {
                    if (picklistValues[j].label != null)
                     {
                        Console.WriteLine(" Item: " + picklistValues[j].label);
                     }
                }
            // Determine whether this field refers to another object
            if (referenceTos != null && referenceTos[0] != null)
            {
                Console.WriteLine("Field references the following objects:");
                for (int j = 0; j < referenceTos.Length; j++)</pre>
                {
                    Console.WriteLine(" " + referenceTos[j]);
               }
          }
       }
   }
}
```

Arguments

The DescribeSObjects () call takes in an array of sObjects.

Name	Туре	Description
sObjectType	string	Object. The specified value must be a valid object for your organization. For a complete list of objects, see <u>Standard Objects</u> .

Response

DescribeSObjectResult

DescribeSObjectResult

The DescribeSObjects () call returns an array of DescribeSObjectResult objects.

describeSoftphoneLayout()

Retrieves layout information for a Salesforce CRM Call Center SoftPhone.

Syntax

```
DescribeSoftphoneLayoutResult[] = binding.describeSoftphoneLayout();
```

Usage

Use this call to obtain information about the layout of a SoftPhone. Use only in the context of Salesforce CRM Call Center; do not call directly from client programs.

Arguments

This call does not take any objects.

Response

The response is a DescribeSoftphoneLayoutResult object:

Name	Туре	Description
CallType	string	A set of attributes that associated with each allowed call type. A call type may be Inbound, Outbound, or Internal.
id	ID	ID of layout. Note that layout objects are not exposed via the API.
name	string	Name of the call type: Inbound, Outbound, or Internal.

CallType

Each describeSoftphoneLayoutResult object contains one or more call types:

Name	Туре	Description
infoFields (may be more than one)	name	The name of an information field in the SoftPhone layout that does not correspond to a Salesforce object. For example, caller ID may be specified in an information field. Information fields hold static information about the call type.
name	string	Name of the layout.
Sections	string	A set of object names and the corresponding item name in the SoftPhone layout, one section for each object in a call type.

Sections

Each call type returned in a describeSoftphoneLayoutResult object contains one section for each call type. Each section contains object-item pairs:

Name	Туре	Description
entityApiName	string	The name of an object in the Salesforce application that corresponds to an item displayed in the SoftPhone layout, for example, a set of accounts or cases.
itemApiName	string	The name of a record in the Salesforce application that corresponds to an item displayed in the SoftPhone layout, for example, the Acme account.

DescribeTabs()

The describeTabs call returns information about the standard and custom apps available to the logged-in user. An app is a group of tabs that works as a unit to provide application functionality. For example, two of the standard Salesforce apps are "Sales" and "Service and Support."

Syntax

DescribeTabSetResult [] = binding.describeTabs();

Usage

Use the DescribeTabs () call to obtain information about the standard and custom apps to which the logged-in user has access. The describeTabs call returns the minimum required metadata that can be used to render apps in another user interface. Typically this call is used by partner applications to render Salesforce data in another user interface.

In the Salesforce user interface, users have access to standard apps (and may also have access to custom apps) as listed in the Force.com app menu at the top of the page. Selecting an app name in the menu allows the user to switch between the listed apps at any time.

For each app, the call returns the app name, the URL of the logo, whether or not it is the currently selected application for the user, and details about the tabs included in that app.

For each tab, the call returns the tab name, the primary SObject4 that is displayed on the tab, whether it is a custom tab, and the URL for viewing that tab. Note that the "All Tabs" tab is never included in the list of tabs.

Sample Code—Java

```
" This tab is selected: " + isSelected);
for (int j=0;j<tabs.length;j++) {
    DescribeTab tab = tabs[j];
    String tabLabel = tab.getLabel();
    String objectName = tab.getSobjectName();
    String tabUrl = tab.getUrl();
    System.out.println("\tTab " + j + 1 + ": \n\t\tLabel = " +
        tabLabel + "\n\t\tObject details on tab: " + objectName + "\n\t\t" +
        "Url to tab: " + tabUrl);
    }
}
catch (Exception ex) {
    System.out.println("\nFailed to describe tabs, error message was: \n" +
        ex.getMessage());
}
```

Sample Code—C#

```
private void describeTabsSample()
{
    sforce.DescribeTabSetResult[] dtsrs = binding.describeTabs();
    Console.WriteLine("There are " + dtsrs.Length.ToString() + " tabsets defined.");
    for (int i=0;i<dtsrs.Length;i++)</pre>
    {
        Console.WriteLine("Tabset " + (i + 1).ToString() + ":");
        sforce.DescribeTabSetResult dtsr = dtsrs[i];
        String tabSetLabel = dtsr.label;
        String logoUrl = dtsr.logoUrl;
        bool isSelected = dtsr.selected;
        DescribeTab[] tabs = dtsr.tabs;
        Console.WriteLine("Label is " + tabSetLabel + " logo url is " + logoUrl + ",
                      there are " + tabs.Length.ToString() + " tabs defined in this set.");
       for (int j=0;j<tabs.Length;j++)</pre>
       {
           sforce.DescribeTab tab = tabs[j];
           String tabLabel = tab.label;
           String objectName = tab.sobjectName;
           String tabUrl = tab.url;
           Console.WriteLine("\tTab " + (j + 1).ToString() + ": \n\t\tLabel = " +
                           tabLabel + "\n\t\tObject details on tab: " + objectName +
             "\n\t\t" + "Url to tab: " + tabUrl);
        }
    }
```

Arguments

None.

Response

DescribeTabSetResult, DescribeTab

DescribeTabSetResult

The DescribeTabs () call returns an array of DescribeTabSetResult objects, which has the following properties:

Name	Туре	Description
label	string	The display label for this standard or custom app. This value changes when tabs are renamed in the Salesforce user interface. See the Salesforce online help for more information.
logoUrl	string	A fully qualified URL to the logo image associated with the standard or custom app.
namespace	string	If this is a custom app, and a set of tabs in the custom app was installed as part of a managed package, the value of this attribute is the developer namespace prefix that the creator of the package chose when the Developer Edition organization was enabled to allow publishing a managed package. This attribute identifies elements of a Force.com AppExchange package.
selected	boolean	If true, then this standard or custom app is the user's currently selected app.
tabs	DescribeTab	An array of tabs that are displayed for the specified standard app or custom app.

DescribeTab

The DescribeTabs () call returns a DescribeTabSetResult object, of which DescribeTab is a property:

Name	Туре	Description
custom	boolean	true if this is a custom tab, false if this is a standard tab.
iconUrl	string	The URL for the main $32 \ge 32$ pixel icon for a tab. This icon appears next to the heading at the top of most pages.
label	string	The display label for this tab.
miniIconUrl	string	The URL for the 16 x 16 pixel icon that represents a tab. This icon appears in related lists and other locations.
sobjectName	string	The name of the SObject4that is primarily displayed on this tab (for tabs that display a particular SObject). For a list of objects, see Standard Objects.
url	string	A fully qualified URL for viewing this tab.

Chapter 11

Utility Calls

This topic describes API calls that your client applications can invoke to obtain the system timestamp, user information, and change user passwords.

Note: For a list of general API calls, see Core Calls. For a list of describe calls, see Describe Calls.

The following table lists the API utility calls described in this topic:

Task / Call	Description
GetServerTimestamp()	Retrieves the current system timestamp from the API.
GetUserInfo()	Retrieves personal information for the user associated with the current session.
ResetPassword()	Changes a user's password to a system-generated value.
SetPassword()	Sets the specified user's password to the specified value.

GetServerTimestamp()

Retrieves the current system timestamp (Coordinated Universal Time (UTC) time zone) from the API.

Syntax

GetServerTimestampResult timestamp = binding.getServerTimestamp();

Usage

Use GetServerTimestamp() to obtain the current system timestamp from the API. You might do this if, for example, you need to use the exact timestamp for timing or data synchronization purposes. When you Create() or Update() an object, the API uses the system timestamp to update the CreatedDate and LastModifiedDate fields, respectively, in the object.

The GetServerTimestamp() call always returns the timestamp in Coordinated Universal Time (UTC) time zone. However, your local system might automatically display the results in your local time based on your time zone settings.



Note: Development tools differ in the way that they handle time data. Some development tools report the local time, while others report only the Coordinated Universal Time (UTC) time zone. To determine how your development tool handles time values, refer to its documentation.

Sample Code—Java

```
public void getServerTimestampSample() {
    // Invoke the getServerTimestamp call and save the results
    try {
        Calendar serverTime = binding.getServerTimestamp().getTimestamp();
        System.out.println("Server time is: " + serverTime.getTime().toString());
    } catch (Exception ex) {
        System.out.println("An unexpected error has occurred." + ex.getMessage());
    }
}
```

Sample Code—C#

```
private void getServerTimeStamp()
{
    //Invoke the getServerTimeStamp call and save the results
    GetServerTimestampResult ts = binding.getServerTimestamp();
    // Write the server timestamp to the diagnostics window
    System.Diagnostics.Trace.WriteLine(ts.timestamp.ToUniversalTime);
}
```

Arguments

None.

Response

GetServerTimestampResult

GetServerTimestampResult

The GetServerTimestamp() call returns a GetServerTimestampResult object, which has the following properties:

Name	Туре	Description
timestamp	dateTime	$System timestamp \ of the \ API \ when \ the \ {\tt GetServerTimestamp} \ () \ call \ was \ executed.$

GetUserInfo()

Retrieves personal information for the user associated with the current session.

Syntax

GetUserInfoResult result = binding.getUserInfo();

Usage

Use GetUserInfo() to obtain personal information about the currently logged-in user. This convenience API call retrieves and aggregates common profile information that your client application can use for display purposes, performing currency calculations, and so on.

The GetUserInfo() call applies only to the username under which your client application has logged in. To retrieve additional personal information not found in the GetUserInfoResult object, you can call Retrieve() on the User object

and pass in the userID returned by this call. To retrieve personal information about other users, you could call Retrieve () (if you know their user ID) or Query () on the User object.

Sample Code—Java

```
public void getUserInfoSample() {
  GetUserInfoResult getUserInfoResult = null;
  try {
    // Invoke the getUserInfo call
    getUserInfoResult = binding.getUserInfo();
   // Display the returned user information
    System.out.println("User's currency symbol: " + getUserInfoResult.getCurrencySymbol());
    System.out.println("User's organization name: " +
                 getUserInfoResult.getOrganizationName());
    System.out.println("User's default currency code: " +
                 getUserInfoResult.getUserDefaultCurrencyIsoCode());
    System.out.println("User's email: " + getUserInfoResult.getUserEmail());
    System.out.println("User's full name: " + getUserInfoResult.getUserFullName());
    System.out.println("User's user id: " + getUserInfoResult.getUserId());
System.out.println("User's language: " + getUserInfoResult.getUserLanguage());
    System.out.println("User's locale: " + getUserInfoResult.getUserLocale());
System.out.println("User's timezone: " + getUserInfoResult.getUserTimeZone());
    System.out.println("User's org is multi currency: " +
                 getUserInfoResult.isOrganizationMultiCurrency());
  } catch (Exception ex) {
    System.out.println("An unexpected error has occurred." + ex.getMessage());
```

Sample Code—C#

```
private void getUserInfo()
{
    //Invoke getUserInfo call and save the results in getUserInfoResult
    GetUserInfoResult ui = binding.getUserInfo();
    // Get some of the user information
    String orgName = ui.organizationName;
    String userFullName = ui.userFullName;
}
```

Arguments

None.

Response

GetUserInfoResult

GetUserInfoResult

The GetUserInfo() call returns a GetUserInfoResult object, which has the following properties:

Name	Туре	Description
accessibilityMode	boolean	Available in API version 7.0 and later. Indicates whether user interface modifications for the visually impaired are on (true) or off (false). The modifications facilitate the use of screen readers such as JAWS.

Name	Туре	Description
currencySymbol	string	Currency symbol to use for displaying currency values. Applicable only when organizationMultiCurrency is false.
userType	string	Type of user license assigned to the Profile associated with the user.
profileID	ID	ID of the profile associated with the role currently assigned to the user.
organizationId	ID	ID of the organization. Allows third-party tools to uniquely identify individual organizations in Salesforce, which is useful for retrieving billing or organization-wide setup information.
organizationMultiCurrency	boolean	Indicates whether the user's organization uses multiple currencies (true) or not (false).
organizationName	string	Name of the user's organization or company.
roleID	ID	Role ID of the role currently assigned to the user.
userDefaultCurrencyIsoCode	string	Default currency ISO code. Applicable only when organizationMultiCurrency is true. When the logged-in user creates any objects that have a currency ISO code, the API uses this currency ISO code if it is not explicitly specified in the Create() call.
userEmail	string	User's email address.
userFullName	string	User's full name.
userID	ID	User ID.
userLanguage	string	User's language, which controls the language for labels displayed in an application. String is 2-5 characters long. The first two characters are always an ISO language code, for example "fr" or "en." If the value is further qualified by country, then the string also has an underscore (_) and another ISO country code, for example "US" or "UK. For example, the string for the United States is "en_US", and the string for French Canadian is "fr_CA."
		For a list of the languages that Salesforce supports, see the Salesforce online help topic "What languages does Salesforce support?"
userLocale	string	User's locale, which controls the formatting of dates and choice of symbols for currency. The first two characters are always an ISO language code, for example "fr" or "en." If the value is further qualified by country, then the string also has an underscore (_) and another ISO country code, for example "US" or "UK. For example, the string for the United States is "en_US", and the string for French Canadian is "fr_CA."
userName	string	User's login name.
userTimeZone	string	User's time zone.
userUiSkin	string	Available in API version 7.0 and later. Returns the value Theme2 if the user is using the newer user interface theme of the online application, labeled "Salesforce." Returns Theme1 if the user is using the older user interface theme, labeled "Salesforce Classic." In the online application, this look and feel setting is configurable at Setup > Customize > User Interface

ResetPassword()

Changes a user's password to a temporary, system-generated value.

Syntax

```
string password = binding.resetPassword(ID userID);
```

Usage

Use ResetPassword () to request that the API change the password of a User or SelfServiceUser, and return a system-generated password string of random letters and numbers. Use SetPassword () instead if you want to set the password to a specific value.

Your client application must be logged in with sufficient access rights to change the password for the specified user. For more information, see Factors that Affect Data Access.

For information on IDs, see ID Field Type.

Sample Code—Java

```
public void resetPasswordSample() {
    // Specify the user ID of the password to reset
    ID idToReset = new ID("005x000000fFLnAAM");
    // Invoke the resetPasswordResult call
    try {
        ResetPasswordResult resetPasswordResult = binding.resetPassword(idToReset);
        // Display the new server-generated password
        System.out.println(resetPasswordResult.getPassword());
    } catch (Exception ex) {
        System.out.println("An unexpected error has occurred." + ex.getMessage());
    }
}
```

Sample Code—C#

```
private void resetPassword()
{
    //Invoke resetPassword call and save results in ResetPasswordResult
    ResetPasswordResult rpr = binding.resetPassword("userID");
    // Get the generated password
    System.Diagnostics.Trace.WriteLine(rpr.password);
}
```

Arguments

Name	Туре	Description
userID	ID	ID of the User or SelfServiceUser whose password you want to reset. For information on IDs, see ID Field Type.

SetPassword()

Sets the specified user's password to the specified value.

Syntax

```
SetPasswordResult setPasswordResult = binding.setPassword(ID userID, string password);
```

Usage

Use SetPassword() to change the password of a User or SelfServiceUser to a value that you specify. For example, a client application might prompt a user to specify a different password, and then invoke SetPassword() to change the user's password. Use ResetPassword() instead if you want to reset the password with a random value generated by the API.

Your client application must be logged in with sufficient access rights to change the password for the specified user. For more information, see Factors that Affect Data Access.

For information on IDs, see ID Field Type.

This call can use the session ID returned in if the password has expired. For more information, see .

Sample Code—Java

```
public void setPasswordSample() {
    // Specify the userID and new password
    ID idToReset = new ID("005x000000fFLnAAM");
    String newPassword = "password";
    try {
        // Invoke the setPassword call
        SetPasswordResult setPasswordResult = binding.setPassword(idToReset, newPassword);
        // If the call fails, an exception is raised; otherwise, the return is empty.
    } catch (Exception ex) {
        System.out.println("An unexpected error has occurred." + ex.getMessage());
    }
}
```

Sample Code—C#

```
private void setPassword()
{
    //Invoke setPassword call; returns nothing if successful
    binding.setPassword("userid", "newpassword");
```

Arguments

Name	Туре	Description
userID	ID	ID of the User or SelfServiceUser whose password you want to reset. For information on IDs, see ID Field Type.
password	string	New password to use for the specified user.

Response

None.

USING THE API WITH SALESFORCE FEATURES

Chapter 12

Implementation Considerations

Before you build an integration application or other client application, consider the data management, use limits, and communication issues explained in this section.

Login Server URL

The API also provides a single login server. You can log in to any organization via a single entry point, without having to hard-code the instance for your organization. To access an organization via the API, you must first authenticate the session by sending a Login() request to the login server at the following URL:

https://www.salesforce.com/services/Soap/c/15.0

The less secure version of the URL is also supported:

```
http://www.salesforce.com/services/Soap/c/15.0
```

The less secure version of the URL is supported for debugging through proxy servers.

Start By Logging In to the Login Server

Before invoking any other calls, a client application must first invoke the Login () call to establish a session with the login server. For more information, see Login () and Step 4: Walk Through the Sample Code.

Salesforce checks the IP address from which the client application is logging in, and blocks logins from unknown IP addresses. For a blocked login via the API, Salesforce returns a login fault. Then, the user must add their security token to the end of their password in order to log in. A security token is an automatically-generated key from Salesforce. For example, if a user's password is mypassword, and their security token is XXXXXXXXX, then the user must enter mypasswordXXXXXXXXX to log in. Users can obtain their security token by changing their password or resetting their security token via the Salesforce user interface. When a user changes their password or resets their security token, Salesforce sends a new security token to the email address on the user's Salesforce record. The security token is valid until a user resets their security token, changes their password, or has their password reset. When the security token is invalid, the user must repeat the login process to log in. To avoid this, the administrator can make sure the client's IP address is added to the organization's list of trusted IP addresses. For more information, see Security Token.

Once you are logged in, you can issue API calls. For each operation, client applications submit a synchronous request to the API, await the response, and process the results. The API commits any changed data automatically. For a list of the API calls:

- Core Calls
- Describe Calls

Typical API Call Sequence

For each call, your client application typically:

- 1. Prepares the request by defining request parameters, if applicable.
- 2. Invokes the call, which passes the request with its parameters to the Force.com Web Service for processing.
- 3. Receives the response from the API.
- **4.** Handles the response, either by processing the returned data (for a successful invocation) or by handling the error (for a failed invocation).

Multiple Instances of Salesforce.com Database Servers

Salesforce.com provides many database server instances. Although organizations are generally allocated by geographic regions, an organization may be on any instance.

Monitoring API Traffic

You can monitor the number of API requests generated by your organization in two ways:

- Any user can see the number of API requests sent in the last 24 hours. To view the information, click Setup ➤ Company Profile ➤ Company Information. Look for the "API Requests, Last 24 Hours" field in the right column.
- If a user has the "Modify All Data" permission, the user can view a report of the API requests sent for the last seven days. To view the information, click the Reports tab, scroll to the Administrative Reports section and click the API Usage Last 7 Days link. Users can sort the report by any of the fields listed in the Summarize Information by: drop-down list. For more information about sorting, filtering, or customizing reports, see the Salesforce online help for reports.

API Usage Metering

To maintain optimum performance and ensure that the API is available to all of our customers, salesforce.com balances transaction loads by limiting the number of API requests (or calls) that any one organization may execute concurrently or within any 24-hour period. The limits are based on edition type, as specified in the following table:

Salesforce Edition		Limit	
•	Developer Edition Trial organizations	•	5,000 calls per organization per 24-hour period, or 5 concurrent calls at any one time

Salesforce Edition	Limit	
 Enterprise Edition Professional Edition with API access enabled 	 1,000 calls per license per 24-hour period, or 1,000,000 calls per organization per 24-hour period, whichever is lower. Minimum rate is 5,000 calls per 24-hour period regardless of the number of licenses. Or, 50 concurrent calls at any one time 	
Unlimited Edition	 5,000 calls per license per 24-hour period, or 5,000,000 calls per organization per 24-hour period, whichever is lower. Or, 50 concurrent calls at any one time 	
Sandbox	 5,000,000 calls per organization per 24-hour period, or 50 concurrent calls at any one time 	

Limits are enforced against the aggregate of all API calls made by the organization in a 24 hour period, and not on a per user basis. When an organization exceeds the limit as calculated above, all users in the organization may be temporarily blocked from making additional calls. Calls will be blocked until enough time has passed for the usage from the preceding 24 hours to drop below the limit. At that time, usage is restored to all users.

In the salesforce.com application, administrators can view how many API requests have been issued in the last 24 hours on the Company Information page at Setup \succ Company Profile \succ Company Information.



Note: Limits are automatically enforced for all editions.

Any action that sends a call to the API counts toward usage limits, except the following:

- Queries from a syndicated feed on a public site
- Outbound messages
- Apex callouts

You can create an API usage metering notification, so that Salesforce will send an email to a designated user when API limits have exceeded a specified limit in a specified time period. For more information, see "About API Usage Notifications" in the Salesforce online help.

There are also limits on the number of requests allowed per organization from the Salesforce user interface. For details, see Concurrent Usage Limits in the Salesforce online help.

Example API Usage Metering Calculations

The following examples illustrate API usage metering calculations for several scenarios:

- For an Enterprise Edition organization with five licenses, the request limit is 5,000 requests (5 licenses X 1000 calls).
- For an Enterprise Edition organization with 15,000 licenses, the request limit is 1,000,000 (the number of licenses X 1000 calls is greater than one million, so the lower limit of one million is used).
- For a Developer Edition organization that made 4,500 calls at 5:00 AM Wednesday, 499 calls at 11:00 PM Wednesday, only one more call could successfully be made until 5:00 AM Thursday.

Compression

Office Toolkit automatically uncompresses responses.

HTTP Persistent Connections

Most clients achieve better performance if they use HTTP 1.1 persistent connection to reuse the socket connection for multiple requests. Persistent connections are handled by the Office Toolkit automatically. For more details, see the HTTP 1.1 specification at:

http://www.w3.org/Protocols/rfc2616/rfc2616-sec8.html#sec8.1

HTTP Chunking

Clients that use HTTP 1.1 may receive chunked responses. Chunking is handled by the Office Toolkit automatically.

Internationalization and Character Sets

The API supports either full Unicode characters or I SO-8859-1 characters. The character set for your organization depends on the Salesforce instance your organization uses. If your organization logs into ssl.salesforce.com, then your encoding is ISO-8859-1. All other instances use UTF-8.

The Office Toolkit submits full Unicode characters to the server. For organizations that use ISO-8895-1 encoding, it is the developer's responsibility to ensure that values are in the correct format. ISO-8859-1 characters in UTF-8 format are not changed.



Note: The API response is encoded in the character set used by your organization (UTF-8 or ISO-8859-1). Either way, the encoded data is usually handled for you by the client.

XML Compliance

The API is based on XML, which requires all documents to be well formed. Part of that requirement is that certain Unicode characters are not allowed in an XML document, even in an escaped form, and that others must be encoded according to their location. Normally this is handled for you by any standard SOAP or XML client. Clients must be able to parse any normal XML escape sequence, and must not pass up invalid XML characters.

Some characters, as mentioned, are illegal even if they are escaped. The illegal characters include the Unicode surrogate blocks and a few other Unicode characters. All are seldom-used control characters that are usually not important in any data, and tend to cause problems with many programs. Although they are not allowed in XML documents, they are allowed in HTML documents and may be present in Salesforce data. The illegal characters will be stripped from any API response.

The following characters are illegal:

- Oxfffe
- Oxffff
- Control characters 0x0 0x19, except the following characters, which are legal: 0x9, 0xA, 0xD, tab, newline, and carriage return)
- 0xD800 0xDFFF

For UTF-8 encoding, Salesforce supports only the basic UCS-2 encoding (two byte, Basic Multilingual Plane), and does not support any of the extended UCS-4 characters. UCS-4 support is extremely rare in any system. UCS-2 is the encoding that Java and Windows support. For more information about XML characters and character sets, see: http://www.w3.org/TR/REC-xml#charsets.

Office Toolkit and Proxies

The Office Toolkit handles proxy settings in the following manner:

- Proxy settings are automatic, or can be set with SetProxyInfo().
- Proxy scripts and explicit proxy settings are supported.
- · Authenticated proxy servers are supported.
- Automatic discovery of proxy servers that are set by a domain controller are not supported.

.Net, Non-String Fields, and the Enterprise WSDL

If you use .Net with the Enterprise WSDL, .Net will generate an extra Boolean field for each non-string field. For example, if you had a date value in MyField_c or the standard field LastModifiedDate, .Net generates a Boolean field for each. In this example, the generated fields would be MyField_cSpecified and LastModifiedSpecified. These field values are False by default. If a Specified field value is False, then the values in the corresponding original field will not be included in the SOAP message.

For example, before the values in the currency field annualRevenue can be included in a SOAP message generated by your client application, the value of annualRevenueSpecified must be set to True:

```
account.annualRevenue = 10000;
account.annualRevenueSpecified = true;
```

Chapter 13

Outbound Messaging

In this chapter ...

- Understanding Outbound Messaging
- Understanding Notifications
- Setting Up Outbound Messaging
- Important Security Considerations
- Understanding the Outbound Messaging WSDL
- Building a Listener

Outbound messaging allows you to specify that changes to fields within Salesforce can cause messages with field values to be sent to designated external servers.

Outbound messaging is part of the workflow rule functionality in Salesforce. Workflow rules watch for specific kinds of field changes and trigger automatic Salesforce actions, such as sending email alerts, creating task records, or sending an outbound message.

Understanding Outbound Messaging

Outbound messaging uses the notifications () call to send SOAP messages over HTTP(S) to a designated endpoint when triggered by a workflow rule.



After you set up outbound messaging, when a triggering event occurs, a message is sent to the specified endpoint URL. The message contains the fields specified when you created the outbound message. Once the endpoint URL receives the message, it can take the information from the message and process it. To do that, you need to examine the outbound messaging WSDL.

Understanding Notifications

A single SOAP message can include up to 100 notifications. Each notification contains the object ID and a reference to the associated SObject4 data. Note that if the information in the object changes after the notification is sent, but before the notification is delivered, only the updated information will be delivered.

If you issue multiple discrete calls, the calls may be batched together into one or more SOAP messages.

Messages will be queued locally. A separate background process performs the actual sending, to preserve message reliability:

- If the endpoint is unavailable, messages will stay in the queue until sent successfully, or until they are 24 hours old. After 24 hours, messages are dropped from the queue.
- If a message cannot be delivered, the interval between retries increases exponentially, up to a maximum of two hours between retries.
- Messages are retried independent of their order in the queue. This may result in messages being delivered out of order.
- You cannot build an audit trail using outbound messaging. While each message should be delivered at least once, it may be delivered more than once. Also, it may not be delivered at all if delivery cannot be done within 24 hours. Finally, as noted above, the source object may change after a notification is sent but before it is delivered, so the endpoint will only receive the latest data, not any intermediate changes.
- Because a message may be delivered more than once, your listener client should check the notification IDs delivered in the notification before processing.



Note: Instead of polling, which was required in previous releases, you can now use outbound messaging to trigger execution logic when Salesforce raises an event. In previous versions of the API, client applications had to poll Salesforce to find out if relevant changes had occurred. Because most changes eventually trigger workflow if a rule exists for it, you can use this to trigger actions based on Salesforce events.

The metadata needed for outbound messaging, including the definition of the notifications () method, which sends the outbound SOAP message to an external service, is in a separate WSDL. The WSDL is created and available from the Salesforce user interface once a workflow rule has been associated with an outbound message. The fields to be delivered in the message are also defined in this WSDL. For more information about setting up outbound messaging, see the Salesforce online help topic "Defining Outbound Messages."

The outbound messaging WSDL is the only WSDL that can be accessed by the API.

Setting Up Outbound Messaging

Before you can use outbound messaging, you must set it up via the Salesforce user interface:

- Setting Up User Profiles
- Defining Outbound Messaging
- Downloading the Salesforce Client Certificate
- Viewing Outbound Messages
- Tracking Outbound Message Status

Setting Up User Profiles

It is possible to create circular changes with outbound messaging. For example, if a user is performing integrations that trigger workflow, and the workflow actions trigger account updates, those account updates trigger new workflow, and so on in an endless loop. In order to prevent these circular changes in Salesforce, you can disable outbound message access for a user.

The following is another example of a circular change scenario:

- 1. You configure an outbound message to include a SessionId on page 343 and specify a user in the User to send as field. The user does not have outbound messaging disabled.
- 2. A change in a contact record triggers an outbound message from the specified user, with the SessionId on page 343 to your outbound message listener.
- 3. Your outbound message listener calls the Force.com API and updates the same contact record which triggered the outbound message.
- 4. The update triggers an outbound message.
- 5. Your outbound message listener updates the record.
- 6. The update triggers an outbound message.
- 7. Your outbound message listener updates the record.

To disable outbound messaging for a user, select "Disable Outbound Messages" in the user's Profile. Salesforce recommends specifying a single user to respond to outbound messages, and disabling outbound messages for this user.

Defining Outbound Messaging

To define outbound messages, use this procedure in the Salesforce user interface:

- 1. Click Setup ➤ Create ➤ Workflow & Approvals ➤ Outbound Messages.
- 2. Click New Outbound Message.
- 3. Choose the object that has the information you want included in the outbound message, and click Next.
- 4. Configure the outbound message.
 - a. Enter a name and description for this outbound message.
 - b. Enter an endpoint URL for the recipient of the message. Salesforce sends a SOAP message to this endpoint.

For security reasons, Salesforce restricts the outbound ports you may specify to one of the following:

- 80: This port only accepts HTTP connections.
- 443: This port only accepts HTTPS connections.
- 7000-10000 (inclusive): These ports accept HTTP or HTTPS connections.
- c. Select the Salesforce user to use when sending the message by specifying a username in the User to send as field. The chosen user controls data visibility for the message that is sent to the endpoint.
- d. Select Include Session ID if you want a Salesforce SessionId to be included in the outbound message. Include the SessionId in your message if you intend to make API calls back to Salesforce from your listener. The SessionId represents the user defined in the previous step and not the user who triggered the workflow.
- e. Select the fields you want included in the outbound message and click Add.
- 5. Click Save.
- 6. On the outbound message detail page, click the Click for WSDL link to view the WSDL associated with this message.

The WSDL is bound to the outbound message and contains the instructions about how to reach the endpoint service and what data is sent to it.



Note: If you do not have these options, your organization does not have outbound messaging enabled. Contact salesforce.com to enable outbound messaging for your organization.

Downloading the Salesforce Client Certificate

Your application (endpoint) server's SSL/TLS may be configured to require client certificates (two-way SSL/TLS), in order to validate the identity of the Salesforce server when it takes the role of client to your server. If this is the case, you can download

the Salesforce client certificate from the Salesforce application user interface. This is the client certificate that Salesforce sends with each outbound message for authentication.

To download the certificate, use this procedure:

- Click Setup ➤ Develop ➤ API to display the WSDL Download page.
- In the WSDL Download page, right-click Download Client Certificate and save it to an appropriate location on your local drive.
- Import the downloaded certificate into your application server, and configure your application server to request the client certificate. The application server then checks that the certificate used in the SSL/TLS handshake matches the one you downloaded.



Note: Your application (endpoint) server must send any intermediate certificates in the certificate chain, and the certificate chain must be in the correct order. The correct order is:

- 1. Server certificate.
- 2. Intermediate certificate that signed the server certificate if the server certificate was not signed directly by a root certificate.
- 3. Intermediate certificate that signed the certificate in step 2.
- 4. Any remaining intermediate certificates. Do not include the root certificate authority certificate. The root certificate is not sent by your server. Salesforce already has its own list of trusted certificates on file, and a certificate in the chain must be signed by one of those root certificate authority certificates.

Viewing Outbound Messages

To view existing outbound messages, click Setup > Create > Workflow & Approvals > Outbound Messages in the Salesforce user interface.

- Click New Outbound Message to define a new outbound message.
- Click View Message Delivery Status to track the status of an outbound message.
- Select an existing outbound message to view details about it or view workflow rules and approval processes that use it.
- Click Edit to make changes to an existing outbound message.
- Click **Del** to delete an outbound message.

Tracking Outbound Message Status

To track the status of an outbound message, click **Setup** > **Monitoring** > **Outbound Messages** in the Salesforce user interface. Alternatively, click **Setup** > **Create** > **Workflow & Approvals** > **Outbound Messages**, and then click **View Message Delivery Status**. From this page you can perform several tasks:

- View the status of your outbound messages including the total number of attempted deliveries
- View the action that triggered the outbound message by clicking any workflow or approval process action ID.
- Click **Retry** to change the **Next Attempt** date to now. This causes the message delivery to be immediately retried.
- Click **Del** to permanently remove the outbound message from the queue.

Important Security Considerations

In order to safely use outbound messaging, you must ensure that no third party can send messages to the endpoint while pretending to be from Salesforce:

- Lock down the client application's listener to accept requests only from the Salesforce IP range. While this guarantees the message came from Salesforce, it does not guarantee that another customer is not pointing to your endpoint and sending messages. The Salesforce IP ranges are:
 - 204.14.232.64 to 204.14.232.79
 - 204.14.234.64 to 204.14.234.79
- Use SSL/TLS. Using SSL/TLS provides confidentiality while data is transported across the internet. Without it, a malicious third party can eavesdrop on your data. This issue is especially important if you pass data with privacy requirements and you pass a SessionId with the message. Also, we authenticate the certificate presented on connection, ensure that it is from a valid Certificate Authority, and check that the domain in the certificate matches the one Salesforce is trying to connect. This prevents us from communicating with the wrong endpoint.
- If the configuration of your application (endpoint) server's SSL/TLS allows, validate the identity of the Salesforce server when it takes the role of a client to your server, using the Salesforce client certificate. For instructions to download the certificate, see Downloading the Salesforce Client Certificate.
- The organization Id is included in each message (see ID Field Type for more information about the Id field type). Your client application should validate that messages contain your organization Id.

Understanding the Outbound Messaging WSDL

The rest of this topic examines relevant sections of the outbound messaging WSDL. Your WSDL may differ, depending on the choices you made when you set up outbound messaging for a particular event on a particular object.

notifications()

This section defines the notifications () call, which creates an outbound message containing specified fields and values for a particular object or objects, and sends the values to a specified endpoint URL:

```
<schema elementFormDefault="qualified" xmlns="http://www.w3.org/2001/XMLSchema"</pre>
 targetNamespace="http://soap.sforce.com/2005/09/outbound">
    <import namespace="urn:enterprise.soap.sforce.com" />
    <import namespace="urn:sobject.enterprise.soap.sforce.com" />
    <element name="notifications">
        <complexType>
            <sequence>
                <element name="OrganizationId" type="ent:ID" />
                <element name="ActionId" type="ent:ID" />
                <element name="SessionId" type="xsd:string" nillable="true" />
                <element name="EnterpriseUrl" type="xsd:string" />
                <element name="PartnerUrl" type="xsd:string" />
                <element name="Notification" maxOccurs="100"</pre>
                  type="tns:OpportunityNotification" />
            </sequence>
        </complexType>
    </element>
</schema>
```

Use this table to understand the elements named in the notifications method definition:

Name	Туре	Description
OrganizationId	ID	ID of the organization sending the message.
ActionId	string	The workflow rule (action) that triggers the message.

Name	Туре	Description
SessionId	string	Optional, a session ID to be used by endpoint URL client that is responding to the outbound message. It is used by the receiving code to make calls back to Salesforce.
EnterpriseURL	string	URL to use to make API calls back to Salesforce using the Enterprise WSDL.
PartnerURL	string	URL to use to make API calls back to Salesforce using the Partner WSDL.
Notification	Notification	Defined in the next section, contains the object datatype and its Id, for example OpportunityNotification or ContactNotification.

The Notification datatype is defined in the WSDL. In the following example, a Notification for opportunities is defined, based on the Notification entry of the notifications () call definition:

Each object element (in our example, opportunities) contains the subset of the fields that you selected when you created the outbound message. Each message Notification also has the object ID, and this needs to be used to track redelivery attempts of notifications you've already processed.

notificationsResponse

This element is the schema for sending an acknowledgement (ack) response to Salesforce.

You acknowledge all notifications in the message if there is more than one.

Building a Listener

Once you have defined an outbound message and configured an outbound messaging endpoint, download the WSDL and create a listener:

- 1. Right-click **Click for WSDL** and select Save As to save the WSDL to a local directory with an appropriate file name. For example, for an outbound message that deals with leads, you could name the WSDL file leads.wsdl.
- 2. Unlike the enterprise or partner WSDLs, which describe the messages the client sends to Salesforce, this WSDL defines the messages that Salesforce will send to your client application.
- 3. Most Web services tools will generate stub listeners for you, in much the same way as they generate a client stub for the enterprise or partner WSDL. Look for a server side stub option.

For example, for .Net 2.0:

- a. Run wsdl.exe/serverInterface leads.wsdl with .Net 2.0. This generates NotificationServiceInterfaces.cs, which defines the notification interface.
- b. Create a class that implements NotificationServiceInterfaces.cs.
- c. You implement your listener by writing a class that implements this interface. There are a number of ways to do this. One simple way is to compile the interface to a DLL first (DLLs need to be in the bin directory in ASP.NET:

```
mkdir bin
csc /t:library /out:bin\nsi.dll NotificationServiceInterfaces.cs
```

Now write an ASMX based Web service that implements this interface. For example, in MyNotificationListener.asmx:

This example is a simple implementation, actual implementations will be more complex.

- **d.** Deploy the service by creating a new virtual directory in IIS for the directory that contains the MyNotificationListener.asmx.
- e. You can now test that the service is deployed by viewing the service page with a browser. For example, if you create a virtual directory salesforce, you'd go to http://localhost/salesforce/MyNotificationListener.asmx.

The process for other Web service tools is similar, please consult the documentation for your Web service tool. For Axis, we recommend version 1.1 or later.

Your listener must meet these requirements:

- Must be reachable from the public Internet.
- For security reasons, Salesforce restricts the outbound ports you may specify to one of the following:
 - 80: This port only accepts HTTP connections.
 - 443: This port only accepts HTTPS connections.
 - 7000-10000 (inclusive): These ports accept HTTP or HTTPS connections.
- To be valid, the common name (CN) of the certificate must match the domain name for your endpoint's server, and the certificate must be issued by a Certificate Authority trusted by Java 2 Platform, Standard Edition (J2SE) 5.0 (JDK 1.5).
- · If your certificate expires, message delivery will fail.



Caution: To avoid an infinite loop of outbound messages that trigger changes that trigger more outbound messages, we recommend that you update objects with a user whose profile has **Disable Outbound Messaging** selected in the Salesforce user interface.

Chapter 14

Data Loading and Integration

In this chapter ...

- Client Application Design
- Salesforce Settings
- Best Practices with Any Data Loader
- Integration and Single Sign-On

If you need to load large volumes of data (hundreds of thousands to millions of records), there are a number of ways you can speed the data loading process. Use the topics in this section to become familiar with issues of client application design, organization configuration, and data loader best practices.

Client Application Design

There are many ways you can design your application to improve the speed of large data loads:

• Client prepares the data efficiently. The client may be reading from a . CSV file, from a database, or some other data source. Client data actions are generally very fast when loading thousands of records. But with large operations every bit of efficiency has a profound effect. If a client is loading a million records, then reducing each row operation by a hundredth of a second would save 55.5 hours. Most client actions are measured in milliseconds already, but if your client does take a half second per record, look for ways to optimize the client.

The Force.com Data Loader is a client application that loads data into or out of any Salesforce object. We recommend using the Data Loader to perform bulk inserts or updates. For more information about the Data Loader, see What is the Data Loader? in the Salesforce online help.

• Use persistent connections. Opening a socket takes time, mostly when opening a socket stems from the SSL/TLS negotiation. (Without SSL or TLS, the API request would not be secure.) Included in the HTTP 1.1 specification is support for reusing sockets among requests (persistent connections) instead of having to re-open a socket per request as in HTTP 1.0. Whether or not your client supports persistent connections depends on the SOAP stack you are using. By default, .Net uses persistent connections, while Apache Axis by default does not. If you change the configuration to use the Apache http-commons libraries, your client will use the HTTP 1.1 specification and persistent connections.

For information about HTTP 1.1, see HTTP Persistent Connections and http://www.w3.org/Protocols/rfc2616/rfc2616-sec8.html#sec8.1.

- Minimize the number of requests. There is some processing associated with each request, so to save time your client should batch as many records per request as possible. Set SetSOAPHeader() to the limit, 2,000. If that is not the most efficient batch size, the API will change the batch size. For more information about setting batch sizes, see SetSOAPHeader().
- Use the fastest internet connection possible. Trying to load a million records over a dial-up modem is not recommended. The API can only load data as fast as you can get the data to it.
- Exercise caution with multi-threaded processes. Some client applications partition large transactions into separate sets of records. Multiple threads run the same integration process on the different data sets. However, do not use multi-threaded processes if you are operating on records that have different parent objects, because anUNABLE_TO_LOCK_ROWS error can occur, causing the entire transaction to fail.

For example, suppose you update contacts every night. You run a multi-threaded process that handled all the contacts beginning with "A" in one thread, and those beginning with "B" in another thread. But one account has a contact beginning with "A" and another with "B", so those two contacts are handled by different threads. When the second thread tries to lock the parent account, it hits a lock already placed by the first thread, and a deadlock occurs. The second thread eventually times out and the transaction fails. Instead, if put every contact of accounts that begin with "A" in one thread and the contacts of accounts that begin with "B" in another thread, there would be no such contention.

Salesforce Settings

Most processing takes place in the database. Setting these parameters correctly will help the database process as quickly as possible:

• Log in as a user with the "Modify All Data" permission to avoid sharing rules. If the client application logs in as a user who has access to data via a sharing rule, then the API must issue an extra query to check access. To avoid this, log in as a user with the "Modify All Data" permission. In general, fewer sharing rules quickens load speeds, as there are fewer operations that have to be performed when setting properties such as ownership.

Alternatively, you can set organization-wide defaults for some objects as public read/write for the duration of the load. For more information, see the Salesforce online help topic "Setting Your Organization Wide Default Sharing Model."

- Avoid workflow or assignment rules. Anything that causes a post-operation action slows down the load. You can temporarily disable automatic rules if the loaded objects are exempt from them.
- Avoid triggering cascading updates. For example, if you update the owner of an account, the contacts and opportunities associated with that account may also require updates. Instead of updating a single object, the client application must access multiple objects, which slows down the load.

Best Practices with Any Data Loader

This section presents a best practice process using the Force.com Data Loader, but the general principals apply to any client data loader:

1. Identify which data you will migrate.

You may not want or need to migrate a whole set of data--choose which objects you wish to migrate. For example, you may want to migrate only the contact information from each account, or only migrate account information from a particular division.

2. Create templates for the data.

Create one template for each object, for example in an Excel worksheet.

Identify the required fields for each object. In addition to the required fields for each standard object, there may be additional required fields such as those needed to follow business rules, or legacy ID fields. Use this guide or see the page layout definitions in the Salesforce user interface to find out which fields are required on standard objects.

You may wish to highlight the required fields in red, to make reviewing the data after you populate the templates easier.

You should also identify any ordering dependencies. Objects may have mandatory relationships, for example all accounts have an owner, and all opportunities are associated with an account. The dependencies in these relationships dictate the order of data migration. For Salesforce data, for example, you should load users first, then accounts, then opportunities.

To identify dependencies, review the related lists and lookup fields in the page layout of the given object, and IDs (foreign keys) in the database.

3. Populate the templates.

Clean your data before populating the template, and review the data in the templates.

4. Migrate the data.

Create custom fields to store legacy ID information. Optionally, give the custom field the External ID attribute so it will be indexed. This will help maintain relationships, and help you build custom reports for validation.

Load one record, check the results, then load all records.

5. Validate the data.

Use all of these techniques to validate your migration:

- · Create custom reports that validate record counts and provide an overall snapshot of migration.
- Spot check the data.
- Review exception reports to see what data was not migrated.
- 6. Re-migrate or update data as needed.

Integration and Single Sign-On



Caution: To avoid getting into an unrecoverable state, do not enable single sign-on for your system administrator account. If you do, and then perform a single sign-on integration that fails, you may not be able to log in again to recover.

Chapter 15

Data Replication

In this chapter ...

- API Calls for Data Replication
- Scope of Data Replication
- Data Replication Steps
- Object-Specific Requirements for Data Replication
- Polling for Changes
- Checking for Structural Changes in the Object

The API supports data replication, which allows you to store and maintain a local, separate copy of your organization's pertinent Salesforce data for specialized uses, such as data warehousing, data mining, custom reporting, analytics, and integration with other applications. Data replication provides you with local control and the ability to run large or ad hoc analytical queries across the entire data set without transmitting all that data across the network.

Use the topics in this section to better understand the best practices for data replication.

API Calls for Data Replication

The API supports data replication with the following API calls:

API Call	Description
GetUpdated()	Retrieves the list of objects that have been updated (added or changed) during the specified timespan for the specified object.
GetDeleted()	Retrieves the list of objects that have been deleted during the specified timespan for the specified object.

Client applications can invoke these API calls to determine which objects in your organization's data have been updated or deleted during a given time period. These API calls return a set of IDs for objects that have been updated (added or changed) or deleted, as well as the timestamp (Coordinated Universal Time (UTC)—not local—timezone) indicating when they were last updated or deleted. It is the responsibility of the client application to process these results and to incorporate the required changes into the local copy of the data.

Scope of Data Replication

This feature provides a mechanism that targets data replication (one-way copying of data). It does not provide data synchronization (two-way copying of data) or data mirroring capabilities.

Data Replication Steps

The following is a typical data replication procedure for an object:

- 1. Optionally, determine whether the structure of the object has changed since the last replication request, as described in Checking for Structural Changes in the Object.
- 2. Call GetUpdated (), passing in the object and timespan for which to retrieve data.

Note that GetUpdated () retrieves the IDs for data to which the logged in user has access. Data that is outside of the user's sharing model is not returned. The API returns the ID of every changed object that is visible to you, regardless of what change occurred in the object. For information on IDs, see ID Field Type.

- 3. Pass in all IDs in an array. For each ID element in the array, call Retrieve() to obtain the latest information you want from the associated object. You must then take the appropriate action on the local data, such as inserting new rows or updating existing ones with the latest information.
- 4. Call GetDeleted(), passing in the object and timespan for which to retrieve data. Like GetUpdated(), GetDeleted() retrieves the IDs for data to which the logged-in user has access. Data that is outside of the user's sharing model is not returned. The API returns the ID of every changed object that is visible to you, regardless of what change occurred in the object. For information on IDs, see ID Field Type.
- 5. Iterate through the returned array of IDs. Your client application must then take the appropriate action on the local data to remove (or flag as deleted) the deleted objects. If your client application cannot match rows in the local data using the retrieved object ID, then the local data rows either were deleted or were never created, in which case there is nothing to do.
- 6. Optionally, save the request time spans for future reference.

Object-Specific Requirements for Data Replication

The API objects have the following requirements for data replication:

- The GetUpdated() and GetDeleted() calls filter the results so that the client application receives IDs for only those created or updated objects to which the logged-in user has access. For information on IDs, see ID Field Type.
- Your client application can replicate any objects to which it has sufficient permissions. For example, to replicate all data for your organization, your client application must be logged in with the "View All Data" permission. For more information, see Factors that Affect Data Access.
- The logged-in user must have read access to the object. For more information, see "Setting Your Organization Wide Default Sharing Model" in the Salesforce online help.
- The object must be configured to be replicateable (Replicateable is true in the SObject4).

Polling for Changes

Client applications typically poll for changed data periodically. Polling involves the following considerations:

- The polling frequency depends on business requirements for how quickly changes in your organization's Salesforce data need to be reflected in the local copy. Some client applications might poll once a day to retrieve changes, while other client applications might poll every five minutes to achieve closer accuracy.
- Deleted records are written to a delete log, which GetDeleted() accesses. A background process that runs every two hours purges records that have been in an organization's delete log for more than two hours if the number of records is above a certain limit. Starting with the oldest records, the process purges delete log entries until the delete log is back below the limit. This is done to protect Salesforce from performance issues related to massive delete logs. The limit is calculated using this formula:

5000 * number of licenses in the organization

For example, an organization with 1,000 licenses could have up to 5,000,000 (five million) records in the delete log before any purging took place. If purging has been performed before your GetDeleted() call is executed, an INVALID REPLICATION DATE error is returned. If you get this exception, you should do a full pull of the table.

• The API truncates the seconds portion of dateTime values. For example, if a client application submits a timespan between 12:30:15 and 12:35:15 (Coordinated Universal Time (UTC) time), then the API retrieves information about items that have changed between 12:30:00 and 12:35:00 (UTC), inclusive.



Note: Development tools differ in the way that they handle time data. Some development tools report the local time, while others report only the Coordinated Universal Time (UTC) time. To determine how your development tool handles time values, refer to its documentation.

- We recommend polling no more frequently than every five minutes. There are built in controls to prevent errant applications from invoking the data replication API calls too frequently.
- Client applications should save the timespan used in previous data replication API calls so that the application knows the last time period for which data replication was successfully completed.
- To ensure data integrity on the local copy of the data, a client application needs to capture all of the relevant changes during polling—even if it requires processing data redundantly to ensure that there are no gaps. Your client application can contain business logic to skip processing objects that have already been integrated into your local data.

- Gaps can also occur if the client application somehow fails to poll the data as expected (for example, due to a hardware crash or network connection failure). Your client application can contain business logic that determines the last successful replication and polls for the next consecutive timespan.
- If for any reason the local data is compromised, your client application might also provide business logic for rebuilding the local data from scratch.

Note: You can now use Outbound Messaging to trigger actions instead of polling for them.

Checking for Structural Changes in the Object

In the API, data replication only reflects changes made to object records. It does not determine whether changes have been made to the structure of objects (for example, fields added to—or removed from—a custom object). It is the responsibility of the client application to check whether the structure of a given object has changed since the last update. Before replicating data, client applications can call DescribeSObjects() on the object, and then compare the data returned in the DescribeSObjectResult with the data returned and saved from previous DescribeSObjects() invocations.

Chapter 16

Feature-Specific Considerations

In this chapter ...

- Archived Activities
- Person Account Record Types
- Opportunity Forecast Override Business Rules
- Call Centers and the API
- Implementing Salesforce Integrations on Force.com

Some Salesforce features require special consideration when accessed via the API. Use the topics in this section to learn about the special considerations for activities, person accounts, forecast override business rules, the Call Center, and creating your own apps.

Archived Activities

Salesforce archives activities (tasks and events) that are over a year old.

You can filter on the isArchived field to find only the archived objects. You cannot use Query () as it automatically filters out all records where isArchived is set to true. You can update or delete archived records, though you cannot update the isArchived field. If you use the API to insert activities that meet the criteria listed below, the activities will be archived during the next run of the archival background process.

Older Events and Tasks are archived according to the criteria listed below. In the Salesforce user interface, users can view archived activities only in the **Printable View** or by clicking **View All** on the Activity History related list or by doing an advanced search.

Person Account Record Types

Beginning with API version 8.0, a new family of record types on Account objects is available: "person account" record types. The person account record types enable specialized business-to-consumer functionality for those who sell to or do business with individuals. For example, a doctor, hairdresser, or real estate agent whose clients are individuals. For more information about person accounts, see "What is a Person Account?" and "Person Account Behaviors" in the Salesforce online help.

Record types are person account record types if the Account field IsPersonAccount is set to true. Salesforce provides one default person account record type, PersonAccount, but an administrator can create additional person account record types. Conversely, record types with the Account field IsPersonAccount set to false are "business account" record types, which are traditional business-to-business (B2B) Salesforce accounts.

When a person account is created (or an existing business account is changed to a person account), a corresponding contact record is also created. This contact record is referred to as a "person contact." The person contact enables the person account to function simultaneously as both an account and a contact. This is the one and only contact record that can be associated directly with the person account. Also, the ID of the corresponding person contact record is stored in the PersonContactId field on the person account.

Review this list of facts about person account record types before working with them:

- You may need to contact your account representative to enable the person account feature.
- You can use a query similar to the following example to find all records with a person account record type:

```
SELECT Name, SobjectType,
IsPersonType FROM RecordType WHERE SobjectType='Account'
AND IsPersonType=True
```

- If you issue a Query() call against an account, the results return the root object type in the SojbectType field. This means that the value returned will always be Account.
- A person contact can be modified, but cannot be created or deleted. Since these kinds of contacts do not have their own record detail page, clients should redirect users to the corresponding person account (Account) page. SOSL results will not include any of the contact fields enabled when IsPersonAccount is set to true. The contact ReportsToId field is not visible.
- If you delete the account, the contact is also deleted. You cannot directly delete the contact; you must delete the account.
- You can change the record type of an account across record type families (typically performed when migrating business accounts to person accounts, but the reverse operation is also supported). When you change the record type from a business account to a person account, the person contact is created. When you change the record type from a person account to a

business account, the person fields are set to null, and the person contact becomes a regular contact with the same parent account it had before the change.



Note: You cannot change record types across record type families in the Salesforce user interface.

- If you change the record type of a business account to a person account using Update(), you cannot make any other changes to fields in that account in the same call; if attempted, an SError will result. However, you can change record type values from one person account record type to another, or from one business account record type to another, in the same call with other changes.
- When converting a business account to a person account, there must be a one-to-one relationship between each business account record and its corresponding contact record. Furthermore, fields common to both records such as Owner and Currency must have identical values.
- Workflow and validation formulas do not fire during a change in record types from or to person accounts. To have workflow or validation formulas trigger when a person account record type is changed to business account (or business account record type is changed to person account), set up a separate trigger that occurs after the update has been made.
- When you change a business account to a person account, valid records will be changed and invalid records will show an error in the results array.
- When you change a person account to a business account, no validation is performed.
- After conversion, the new person accounts will have unique one-to-one relationships with the contact records that formed them. As is true for all person accounts, no other contacts can be associated to a person account.
- After conversion, any existing account field history information remains on the person accounts. Any existing contact field history information is retained on the contact, but is not added to the person accounts field history.

For more information about person accounts, see the Salesforce online help.

Opportunity Forecast Override Business Rules

Customizable forecasting is the preferred way to track revenue data in Salesforce. If you have customizable forecasting enabled for your organization, users with the "Override Forecast" permission can override forecast amounts for themselves and their direct reports. For more information, see "Overriding Customizable Forecasts" in the Salesforce online help.

You can use the following objects to retrieve opportunity forecast override information:

- OpportunityOverride
- LineitemOverride

These read-only objects keep an up-to-date record of the current values related to forecasts, whether those values are inherited directly from the opportunity or reflect an opportunity forecast override. Inherited values are stored in these objects, separate from the opportunity object, to improve performance.

If an override record exists (whether or not a particular value is overridden), the Amount, Quantity, Forecast Period, Forecast Category values. and Unit Price on LineitemOverride values stored in OpportunityOverride or LineitemOverride are used to compute the forecast for the OwnerId specified in the record. These objects represent the user's view of the Opportunity or OpportunityLineItem with regard to forecasting, and may reflect overridden values from an opportunity owner or Forecast Manager below the specified user in the Forecast Hierarchy. If an override record does not exist, then the values stored in the Opportunity or OpportunityLineItem record are used, and no override information appears to the user.

In the Forecast Hierarchy, one user at each non-leaf level is designated as the Forecast Manager. The Forecast Manager sees the Opportunity Forecast Related List on the Opportunity Detail page when they view an opportunity owned by a subordinate user (if that user has the Allow Forecasting permission). A Forecast Manager's forecasts include contributions from those
same subordinates. Other users in the same role as a Forecast Manager just see their own opportunities reflected in their forecasts.

OpportunityOverride Lifecycle

OpportunityOverride records are created, updated, or deleted only under certain conditions, and only for relevant users.



Note: Because LineitemOverride records always have a parent record in the OpportunityOverride object with the same OpportunityId and OwnerId, this section explains the lifecycle of the OpportunityOverride records first, then explains any additional behaviors relevant only to LineitemOverride records.

Creating Opportunity Overrides

Forecast managers can override a forecast-related value (such as Amount, Quantity, Forecast Period, and ForecastCategory) by editing the Opportunity Forecasts related list on an opportunity owned by a subordinate user. Also, the opportunity owner can change Opportunity forecast-related values (such as ForecastCategory, StageName, Amount, and CloseDate) on an Opportunity that he or she owns. ForecastCategory has additional rules governing it; see Special Case: Forecast Category for more information.

When a user sets the first override for a particular Opportunity record, OpportunityOverride records are created and appropriate values are written for the opportunity owner and every forecast manager above the Opportunity owner in the forecast hierarchy. The values stored in each user's record depend on their location in the hierarchy—users below the user who made an override will have the original values. Other users will have the overrides, plus the Opportunity values for anything not overridden.

When subsequent override values are set, the new values are written to the OpportunityOverride records of the user who set the override and other forecast managers above in the forecast hierarchy, until an existing override value is reached.

Quantity can only be overridden in the Salesforce user interface if Quantity Forecasting is enabled on the Forecast Settings page, and Amount can only be overridden if Revenue Forecasting is enabled.



Note: Any changes that an opportunity owner can make on the Opportunity Forecast edit page are applied to the Opportunity record as well as the relevant OpportunityOverride records.

Updating Opportunity Values and Opportunity Override Values

Whenever an opportunity forecast-related value is updated on the Opportunity record itself, the corresponding records in the OpportunityOverride for the Opportunity owner and forecast managers above the owner in the forecast hierarchy are updated, until the first override value is encountered. The Opportunity owner cannot override anything except ForecastCategory, but he or she can edit the CloseDate and StageName on the Opportunity Forecast edit page, in addition to ForecastCategory. All three fields correspond to the Opportunity record, and the result is the same whether they make the changes on the Opportunity Forecast edit page.

Changes to the following Opportunity fields also affect records in the OpportunityOverride:

- CloseDate affects the forecast PeriodId if the new CloseDate falls into a different Forecast Period than the old CloseDate. In this case, the OpportunityOverride PeriodId is updated on the opportunity override record of the opportunity owner, and all forecast managers above the owner in the forecast hierarchy, until the first overriden PeriodId is encountered.
- CurrencyType IsoCode changes always cause an update to the opportunity owner's OpportunityOverride. No other object is updated unless Amount changes, which is often the case.
- If territory management is enabled, territory changes on the Opportunity affect OpportunityOverride records. An Opportunity owner's OpportunityOverrides are updated with the new territory, and OpportunityOverride records are inserted or updated for forecast managers above the Opportunity owner in the forecast hierarchy. OpportunityOverride records are also deleted for any forecast manager whose territory (stored on the override record) is no longer above the Opportunity owner in the new opportunity owner remains a subordinate of a

forecast manager before and after the update, if they roll up into a different territory, that forecast manager loses the OpportunityOverrides for the old territory and new records are created for the new territory.

- When an Opportunity is set to Closed Won (the StageName value is Closed Won), the AmountInherited, QuantityInherited, and PeriodInherited fields in OpportunityOverride are set to true and the corresponding values are updated to match the Opportunity, if they are different. The OpportunityOverride ForecastCategoryInherited flag is also set to true and the Opportunity ForecastCategory is set to Closed, unless it has been overridden to Omitted. Omitted is the only valid ForecastCategory override for a Closed Won opportunity.
- When an Opportunity is set to Closed Lost, the Amount and PeriodInherited flags are set to true, and the corresponding values are updated to match the Opportunity, if they are different, and the ForecastCategoryInherited flag is set to true and the ForecastCategory is set to Omitted if the current value is anything other than Omitted.

Deleting Opportunity Overrides

OpportunityOverride records are not deleted unless one of the following triggering events occurs:

- An Opportunity is deleted.
- A change is made in the forecast hierarchy, such that a particular OpportunityOverride owner is no longer above the Opportunity owner or is no longer the forecast manager in the relevant role or territory.
- An Opportunity is transferred to a new owner or territory, individually or as part of an account transfer.
- The "Allow Forecasting" permission is removed from a user.

When an Opportunity is transferred to a new owner, an OpportunityOverride record for the new owner is added as needed, and OpportunityOverride records are deleted for all users that are not forecast managers above the new owner in the forecast hierarchy, possibly including the previous Opportunity owner. Also, OpportunityOverride records are updated or inserted as necessary for all forecast managers above the new owner in the forecast hierarchy. Restrictions on Opportunity owners are strictly enforced during a transfer. If the new owner was in the forecast hierarchy of the previous owner, and had made overrides that are no longer valid as the opportunity owner, such as an Amount override, then the corresponding inherited flag is set to true and the value is refreshed from the opportunity.

Special Case: Forecast Category

An additional rule applies to ForecastCategory values: If you set the ForecastCategory in a new opportunity that is not the default forecast category for the stage selected, or update an existing ForecastCategory on an opportunity, then OpportunityOverride records are created for relevant users as described above. Since only Opportunity owners can update the ForecastCategory on an opportunity record directly, this scenario is treated as a ForecastCategory override by the opportunity owner (the ForecastCategoryInherited flag on that user's OpportunityOverride record is false).

LineitemOverride Object Lifecycle

Additional rules apply to line item overrides, as explained in the rest of this section.

The full set of LineitemOverride records for each OpportunityOverride record is always created to mirror the line item records that exist for the Opportunity record. Whether a user makes an override at the opportunity level or line item level, or even just edits Forecast Category on the Opportunity Edit (only opportunity owners can do this), the end result is the same in terms of the Opportunity and LineitemOverride records that are created.

All information from Updating Opportunity Values and Opportunity Override Values applies to updating LineitemOverride records, with these changes:

• Forecast Category overrides at the opportunity level are cascaded down into child LineitemOverride records, if the Forecast Category has not been overridden for a particular line item by that user or a subordinate user. That is, when a user establishes an opportunity-level Forecast Category override, any line item-level Forecast Category overrides by a subordinate forecast manager will take precedence when updating line item Forecast Category values for this user and any superiors. The converse is not true. Forecast Category on an OpportunityOverride record is never updated in response to a Forecast

Category override on a LineitemOverride record by any user. When a user sets a Forecast Category override at the line item level, the override value is written to the LineitemOverride records of the user who set the override and other forecast managers above that user in the forecast hierarchy, until an existing override value is reached.

• Opportunity line item values for Unit Price, Total Price and Quantity, if changed, are saved on the Lineitem Override records for the opportunity owner and above in the forecast hierarchy, until the first overriden value is encountered for the field. Quantity can only be overridden in the Salesforce user interface if Quantity Forecasting is enabled on the Forecast Settings page. Unit Price and Total Price can only be overridden if Revenue Forecasting is enabled.

When Quantity or Unit Price are overridden, the Total Price is computed and written to the relevant LineitemOverride record. When Total Price is overridden, but not Unit Price, the Unit Price is computed and written to the relevant LineitemOverride records. If both are overridden, no computation occurs. The computation applies to the LineitemOverride record that contains the override value and related LineitemOverride records that inherit the overridden value. That is, for a given LineitemOverride record, we compute Total Price or Unit Price according to the above rules, if an override is made on the current record or if the record inherited an override made by a subordinate forecast manager.

Note that if you have selected the value Schedule Date in the Setup > Customize > Forecast > Forecast Settings > Forecast Date picklist, you cannot override these values:

- If forecasting by Revenue, or Revenue and Quantity, and the OpportunityLineItem record has a Revenue schedule, you cannot override Unit Price and Total Price.
- If forecasting by Quantity, or Revenue and Quantity, and the OpportunityLineItem records have Quantity schedules, you cannot override Quantity.

Whenever a line item is deleted, LineitemOverride records relating to that line item are deleted for all users, similar to opportunities. Also, opportunity transfers and forecast hierarchy changes affect line item overrides the same way they affect opportunity overrides. That is, if an OpportunityOverride record is deleted, all the child LineitemOverride records, matching on OpportunityId and OwnerId, are also deleted.

Call Centers and the API

The API provides access to information about computer telephony integration (CTI) call centers with the describeSoftphoneLayout () call. You must have the CTI Integration feature enabled for your organization. Contact your account representative for assistance.

The API supports limited access to call center-related objects, including being able to create call centers, and create or modify additional numbers for the call center:

Торіс	Description
CallCenter	Call Center object description, including fields and usage.
AdditionalNumber	Configuration settings that allow you to add an additional number if it cannot easily be categorized as a user, contact, lead, account, or any other object. Examples include phone queues or conference rooms.

In addition, several fields have been added to existing objects to support call centers. The following fields provide configuration settings for operation of a call center:

Object Name	Field Name	Field Type	Field Properties	Description
OpenActivity ActivityHistory	CallDisposition	string	Create (Task only)	Represents the result of a given call, for example, "we'll call back," or "call unsuccessful." Limit is 255 characters. For the Task object, corresponds to the Salesforce user interface label Call Result . You can also create and update values for this field in Task.
Task			Filter Nillable	
			<mark>Update</mark> (Task only)	
OpenActivity	CallDurationInSeconds	int Creat (Task	Create (Task	Duration of the call in seconds. For Task, you can also create and update values for this field.
ActivityHistory			only)	
Task			Filter	
			Update (Task only)	
OpenActivity	CallObject	string	Filter	Name of a call center. Limit is 255 characters.
ActivityHistory			Nillable	For Task, you can also create and update values for this field.
Task			Update (Task only)	
OpenActivity ActivityHistory	CallType	picklist	Create (Task only)	The type of call being answered: Inbound, Internal, or Outbound.
Task			Filter	For Task, you can also create and update values for this field.
			Nillable	
			Restricted picklist	
			Update	
User	CallCenterId	reference	Create	The unique identifier for the call center associated with this user.
			Filter	
			Nillable	
			Update	
User	UserPermissionsCall CenterAutoLogin	boolean	Create	Indicates whether a user will be automatically logged in to a call center when logging in to the Salesforce application (true) or not (false).
			Update	

Implementing Salesforce Integrations on Force.com

You can implement your Salesforce integrations or other client applications, on the Force.com platform by creating a Force.com AppExchange app:

1. Create a WebLink that passes the user session ID and the API server URL to an external site:

https://www.your_tool.com/test.jsp?sessionid={!API_Session_ID}&url={!API_Partner_Server_URL_80}

Use https to ensure your session ID cannot be detected.

- 2. The page pointed to in the step above takes the session ID and uses it to call back to the API. Use GetUserInfo() to return the userID associated with the session and related information. If needed, you can also use retrieve on the User object to retrieve any additional information you need about the user.
- 3. Maintain a cross-reference between the UserId or username and the corresponding user ID in your system, which you can do using a WebLink that is executed when the user clicks on a tab, or a WebLink on the page layout.
- 4. Package and upload this app using the instructions in the Salesforce online help topic "Preparing Your Apps for Distribution."

Index

.Net handling of non-string fields 447

A

Account object 67 AccountContactRole object 75 AccountHistory object 76 AccountOwnerSharingRule object 76 AccountPartner object 78 AccountShare object 79 AccountTag object 81 AccountTeamMember object 82 AccountTerritoryAssignmentRule object 83 AccountTerritoryAssignmentRuleItem object 84 ActivityHistory 87 AdditionalNumber object 89 anti-join 376 Apex SOSL FIND clause syntax 407 ApexClass object 91 ApexComponent object 92 ApexPage object 94 ApexTrigger object 95 API usage metering 444 Approval object 96 Asset object 98 AssetTag object 101 AssignmentRule object 101 AsyncApexJob object 102 asynchronous API calls 40 Attachment object 103 authentication 46

B

Bookmark object 105 BrandTemplate 106 BusinessProcess object 110

C

Call centers and the API 469 CallCenter object 111 calls 363 convertLead 354 Create 359 CreateObject 362 describeSObject 427 describeSObjects 429 describeSoftphoneLayout 432 describeTabs 433 GetDeleted 363 getServerTimestamp 436 GetUpdated 365 getUserInfo 437 list of core calls 354 list of describe calls 426 list of utility calls 436 Login 368 Query 369 Refresh 398

calls (continued) resetPassword 440 Retrieve 399 Search 401 setPassword 441 SetServerURL 419 SetSOAPHeader 420 undelete 421 Update 422 Campaign object 112 CampaignMember object 117 CampaignMemberStatus object 118 CampaignOwnerSharingRule object 120 CampaignShare object 120 CampaignTag object 121 cascading deletes 33 Case object 122 CaseComment object 126 CaseContactRole object 127 CaseHistory object 128 CaseOwnerSharingRule object 129 CaseShare object 130 CaseSolution object 131 CaseStatus object 132 CaseTag object 133 CaseTeamMember object 134 CaseTeamRole object 134 CaseTeamTemplate object 135 CaseTeamTemplateMember object 135 CaseTeamTemplateRecord object 136 CategoryData object 136 CategoryNode object 137 CategoryNodeLocalization object 138 Changes 12 characters reserved in SOQL SELECT 380 characters reserved in SOSL FIND 407 child relationships identifying 393 ChildRelationship4 object 327 chunking of HTTP 446 cipher key length 46 client certificate download 451 Community object 140 comparison operators 375 compression 446 condition expression (WHERE clause) 374 connections, HTTP persistent 446 Contact object 141 ContactHistory object 145 ContactOwnerSharingRule object 146 ContactShare object 147 ContactTag object 148 Contract object 149 ContractContactRole object 153 ContractHistory object 154 ContractStatus object 154 ContractTag object 155 Conventions **SOQL 372** SOSL 404 convertLead calls 354 count() 380

Create call 359 CreatedById fields 29 CreatedDate fields 29 CreateObject call 362 Currency fields, querying with SOSL 416 CurrencyType object 156 custom objects 34

D

data types, mapping API data types to Salesforce field types 28 database server instances 444 date formats 384 date literals 384 DatedConversionRate object 157 deleting and cascading deletes 33 describeSObject call 427 describeSObjects call 429 describeSoftphoneLayout call 432 describeTabs call 433 Division object 158 Divisions, filtering in SOSL 415 Document object 159 DocumentAttachmentMap 162 DocumentTag object 163

E

effective dated currency 157 EmailMessage object 163 EmailServicesAddress object 165 EmailServicesFunction object 166 EmailStatus 170 EmailTemplate object 171 DocumentAttachmentMap 162 Enterprise WSDL and .Net 447 EntityHistory object 173 error handling 43 escape sequences for quoted string 379 Event object 174 EventAttendee object 181 EventTag object 182 expiration of session ID 44

F

feed service API URL Syntax 397 field expression 375 field types 23 Field4 object 327 fields .Net and the Enterprise WSDL handling of non-string fields 447 system fields 29 FIND clause, SOSL 405 syntax used in Apex 407 FiscalYearSettings object 183 Folder object 185 ForecastShare object 186 foreign keys null value and relationship query 392

G

GetDeleted call 363 getServerTimestamp call 436 GetUpdated call 365 getUserInfo call 437 Group object 187 GroupMember object 188

Η

Holiday object 189 HTTP chunking 446 HTTP persistent connections 446

Ι

ID fields 29 Idea object 192 IdeaComment object 194 implicit security restrictions for objects and field 48 IN clause, SOSL 408 internationalization 446 ISO-8859-1 446

L

LastModifiedById fields 29 LastModifiedDate fields 29 Lead object 195 LeadHistory object 201 LeadOwnerSharingRule object 202 LeadShare object 202 LeadStatus object 203 LeadTag object 204 LIMIT 381 LIMIT clause, SOSL 415 limiting API requests (API usage metering) 444 LineItemOverride object 205 localization and character sets 446 LocalizationContext4 object 332 logical operators 379 login 46 Login call 368 login server URL 443 lookup relationships 33

\mathbf{M}

MailMergeTemplate object 206 mapping API data types to Salesforce field types 28 master-detail relationships 33 MergeRequest8 object 333 metering API usage 444 Multicurrency organizations, querying currency fields with SOSL 416

Ν

Note object 209 NoteAndAttachment object 210 NoteTag object 210 notifications 450 notifications call in outbound messaging WSDL 453 null values in foreign key with relationship query 392

0

objects Account 67 AccountContractRole 75

Index

objects (continued) AccountHistory 76 AccountOwnerSharingRule 76 AccountPartner 78 AccountShare 79 AccountTag 81 AccountTeamMember 82 AccountTerritoryAssignmentRule 83 AccountTerritoryAssignmentRuleItem 84 ActivityHistory 87 AdditionalNumber 89 ApexClass 91 ApexComponent 92 ApexPage 94 ApexTrigger 95 Approval 96 Asset 98 AssetTag 101 AssignmentRule 101 AsyncApexJob 102 Attachment 103 Bookmark 105 BrandTemplate 106 BusinessProcess 110 CallCenter 111 Campaign 112 CampaignMember 117 CampaignMemberStatus 118 CampaignOwnerSharingRule 120 CampaignShare 120 CampaignTag 121 Case 122 CaseComment 126 CaseContactRole 127 CaseHistory 128 CaseOwnerSharingRule 129 CaseShare 130 CaseSolution 131 CaseStatus 132 CaseTag 133 CaseTeamMember 134 CaseTeamRole 134 CaseTeamTemplate 135 CaseTeamTemplateMember 135 CaseTeamTemplateRecord 136 CategoryData 136 CategoryNode 137 CategoryNodeLocalization 138 community 140 Contact 141 ContactHistory 145 ContactOwnerSharingRule 146 ContactShare 147 ContactTag 148 Contract 149 ContractContactRole 153 ContractHistory 154 ContractStatus 154 ContractTag 155 CurrencyType 156 custom objects 34 DatedConversionRate 157 Division 158 Document 159 DocumentAttachmentMap 162 DocumentTag 163 EmailMessage 163 EmailServicesAddress 165

objects (continued) EmailServicesFunction 166 EmailStatus 170 EmailTemplate 171 EntityHistory 173 Event 174 EventAttendee 181 EventTag 182 FiscalYearSettings 183 Folder 185 ForecastShare 186 Group 187 GroupMember 188 Holiday 189 idea 192 idea comment 194 Lead 195 LeadHistory 201 LeadOwnerSharingRule 202 LeadShare 202 LeadStatus 203 LeadTag 204 LineItemOverride 205 MailMergeTemplate 206 Note 209 NoteAndAttachment 210 NoteTag 210 **OpenActivity 211 Opportunity 214** OpportunityCompetitor 219 OpportunityContactRole 220 OpportunityFieldHistory 221 **OpportunityHistory 221** OpportunityLineItem 223 OpportunityLineItemSchedule 226 OpportunityOverride 229 OpportunityOwnerSharingRule 230 **OpportunityPartner 231 OpportunityShare 231 OpportunityStage 233** OpportunityTag 235 OpportunityTeamMember 234 OrderTag 240 Partner 250 PartnerNetworkConnection 251 PartnerNetworkRecordConnection 252 PartnerRole 255 Period 255 Pricebook2 256 PricebookEntry 258 ProcessInstance 260 ProcessInstanceHistory 261 ProcessInstanceStep 262 ProcessInstanceWorkitem 263 Product2 264 Profile 268 QuantityForecast 270 QuantityForecastHistory 274 QueueSobject 276 RecordType 276 RevenueForecast 278 RevenueForecastHistory 282 Scontrol 284 ScontrolLocalization 286 SelfServiceUser 287 Solution 289 SolutionHistory 291 SolutionStatus 291

objects (continued) SolutionTag 292 standard objects 60 StaticResource 293 TagDefinition 295 Task 296 TaskPriority 299 TaskStatus 300 TaskTag 301 Territory 302 User 304 UserAccountTeamMember 312 UserLicense 314 UserPreference 315 UserRole 316 UserTeamMember 318 UserTerritory 319 vote 320 WebLink 321 WebLinkLocalization 324 OpenActivity object 211 operators comparison 375 logical 379 Opportunity object 214 OpportunityCompetitor object 219 OpportunityContactRole object 220 OpportunityFieldHistory object 221 OpportunityHistory object 221 OpportunityLineItem object 223 OpportunityLineItemSchedule object 226 OpportunityOverride object 229 OpportunityOwnerSharingRule object 230 OpportunityPartner object 231 OpportunityShare object 231 OpportunityStage object 233 OpportunityTag object 235 OpportunityTeamMember object 234 ORDER BY 380 ORDER BY clause, SOSL 414 OrderTag object 240 outbound messaging and notifications 450 building a listener 454 configuring up user profiles 450 defining 451 introduction 448 notification calls in WSDL 453 port restrictions 451 security 452 setting up 450 tracking 452 understanding 449 viewing 452

P

package API access 48 parent relationships identifying 393 Partner object 250 PartnerNetworkConnection object 251 PartnerNetworkRecordConnection object 252 PartnerRole object 255 Period object 255 persistent connections 446 picklists multi-select picklists 432 picklists (continued) picklistValues 327, 432 restrictedPicklist 327, 432 single-select picklists 432 PickListValue4 object 333 port restrictions for outbound ports 50 Pricebook2 object 256 PricebookEntry object 258 ProcessInstance object 260 ProcessInstanceHistory object 261 ProcessInstanceStep object 262 ProcessInstanceWorkitem 263 Product2 object 264 Profile object 268

Q

QuantityForecast object 270 QuantityForecastHistory object 274 query SOQL overview 372 Query call 369 QueryResultSet4 object 334 QueueSobject object 276 quoted string escape sequences 379

R

RecordType object 276 Refresh call 398 relationship names 388 for custom objects and custom fields 390 relationship queries 388 and foreign key with null values 392 history object 396 identifying parent and child relationships 393 limitations 395 overview 389 partner WSDL 396 polymorphic keys 395 understanding results 392 reserved characters in SOQL SELECT 380 reserved characters in SOSL FIND 407 resetPassword call 440 Retrieve call 399 **RETURNING clause, SOSL 409** RevenueForecast object 278 RevenueForecastHistory object 282 Role, see UserRole 316

S

salesforce.com instances 444 SBriefcaseType object 334 Scontrol object 284 ScontrolLocalization object 286 search SOSL overview 402 Search call 401 security 45 and packages 48 implicit restrictions for objects and fields 48 outbound messaging 452 outbound port restrictions 50 sharing 47 user profile configuration 46 security token 46

count() 380 date formats 374 date formats and date literals 384 field expression 375 LIMIT 381 ORDER BY 380 SOQL SELECT 373 toLabel() 383 SelfServiceUser object 287 semi-join 376 SError object 335 session ID expiration 44 setPassword call 441 SetServerURL call 419 SetSOAPHeader call 420 SForceSession4 object 341 SforceSessionEvents4 object 346 sharing 47 SObject4 object 347 Solution object 289 SolutionHistory object 291 SolutionStatus object 291 SolutionTag object 292 SOQ-R, see relationship queries 388 SOQL compared with SOSL 403 overview 372 reserved characters 380 typographical conventions 372 SOQL relationship queries 388 SOSL about 403 compared with SOQL 403 examples 417 FIND clause 405 IN clause 408 LIMIT clause 415 ORDER BY clause 414 overview 402 querying currency fields 416 **RETURNING** clause 409 syntax 404 syntax used in Apex 407 test search in CJK languages 418 toLabel() call 416 typographical conventions 404 WHERE clause 411 WITH clause 415 SSL requirement 46 standard objects 60 StaticResource object 293 support policy backward compatibility 11 call deprecation 11 deprecated calls 11

Office Toolkit 11

synchronous API calls 40 system fields 29 SystemModstamp fields 29

Т

Tab4 object 350 TabSet4 object 350 TabSetCollection4 object 351 TagDefinition object 295 Task object 296 TaskPriority object 299 TaskStatus object 300 TaskTag object 301 Territory object 302 TLS requirement 46 types of fields 23 Typographical conventions SOQL 372 SOSL 404

U

undelete 421 Unicode 446 Update call 422 URL syntax for feed service API 397 user authentication 46 User object 304 user profile configuration 46 user profiles configuring up for outbound messaging 450 UserAccountTeamMember object 312 UserInfoResult8 object 352 UserLicense object 314 UserPreference object 315 UserRole object 316 UserTeamMember object 318 UserTerritory object 319 UTF-8 446

V

Vote object 320

W

WebLink object 321 WebLinkLocalization object 324 What's New 12 WHERE clause, SOSL 411 WITH clause, SOSL 415

Х

XML compliance 446