Rebate Management Developer Guide

Salesforce, Summer '25

Last updated: August 8, 2025

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

CONTENTS

Rebate Management Standard Objects 1

AccountAccountRelation 3
PartyRoleRelation 6
ProgramRebateType 8
ProgramRebateTypeBenefit 14
PgmRebateTypBnftMapping 16
ProgramRebateTypeFilter 18
ProgramRebateTypePayout 20
ProgramRebateTypPayoutSrc 21
ProgramRebateTypeProduct 24
ProgramRebateTypeReference 25
RebateClaim 26
RebateClaimAdjustment 34
RebateMemberAggregateItem 36
RebateMemberClaimAggregate 39
RebateMemberProductAggregate 42
RebatePartnerSpecialPrcTrm 46
RebatePayment 50
RebatePayoutAdjustment 51
RebateProgram 53
RebateProgramMember 55
RebateProgramMemberPayout 58
RebateProgramPayoutPeriod 60
RebatePtnrSpclPrcTrmBnft 62
ReceivedDocument 65
TransactionJournal 66
UnitOfMeasureConversion 69

Rebate Management Standard Invocable Actions 71

Add Rebate Member List 71
Calculate Rebate Amount and Upsert Payout 73
Calculate Projected Rebate Amount 74
Generate Rebate Payout Periods 77
Get Benefit and Calculate Rebate Amount 79
Get Eligible Program Rebate Types 81
Process Program Rebate Type Products 84
Process Rebate Batch Calculation Jobs 85
Process Rebate CSV Files 87
Upsert Custom Rebate Payout 88

Rebate Management Metadata API Types 91
ObjectHierarchyRelationship 91
Data Processing Engine, Batch Management, and Monitor Workflow Services 97
Data Model 98
Common Tooling API Object 99
Common Platform Event 105
Common Business APIs 107
Data Processing Engine 109
Batch Management 148
Monitor Workflow Services 166

REBATE MANAGEMENT STANDARD OBJECTS

Rebate Management data model provides objects and fields to create and manage rebate programs and manage payouts and transactions. Use the objects to optimize the incentives offered to build a mutually profitable relationship with your partners.

Rebate Management is available in Lightning Experience.

Available in: Enterprise, Unlimited, and Developer Editions.

AccountAccountRelation

Represents a relationship between accounts, such as a relationship between a distributor account and a ship-to account. This object is available in API version 58.0 and later.

PartyRoleRelation

Represents information about the type of relationship between the participants in a relationship. This object is available in API version 58.0 and later.

ProgramRebateType

Provide the rebate types that are part of this program. For example, volume rebate, revenue rebate, or rebate on every transaction. This object is available in API version 51.0 and later.

ProgramRebateTypeBenefit

Defines the benefit matrix for the rebate type. For example, 5% or $200. This object is available in API version 51.0 and later.

PgmRebateTypBnftMapping

If benefit table is extended, defines mapping of benefit field to the aggregate object fields. This object is available in API version 51.0 and later.

ProgramRebateTypeFilter

The definition that filters the transaction journals eligible for a rebate type. This filter definition is used in the rebates data processing engines. This object is available in API version 51.0 and later.

ProgramRebateTypePayout

The payout given to a member for a particular rebate type. For example, volume rebate payout in Jan'19 for ABC enterprises is $560, petrol engine payout for ABC in Jan'19 is $440. This object is available in API version 51.0 and later.

ProgramRebateTypPayoutSrc

The rebate amount and the tier applied calculated for each row in the aggregate. There is a 1 to 1 relation between payout source and aggregate. This object is available in API version 51.0 and later.

ProgramRebateTypeProduct

Represents a junction between a program rebate type and a product. This object is available in API version 53.0 and later.

ProgramRebateTypeReference

Represents the association between the contract, opportunity, or any eligible standard or custom object, and rebate type. This object is available in API version 52.0 and later.

RebateClaim

Represents information about the claim submitted by the end customer or distributor for a ship and debit program. This object is available in API version 54.0 and later.

RebateClaimAdjustment

Represents information about the adjustments made to the rebate claim. This object is available in API version 58.0 and later.

RebateMemberAggregateItem

Represents a junction between a rebate member product aggregate and a transaction journal. This object is available in API version 54.0 and later.

RebateMemberClaimAggregate

Represents information about the aggregated claim quantity and amount for a rebate member per product. This object is available in API version 58.0 and later.

RebateMemberProductAggregate

Stores the post calculation summary of journal transactions by member, period, and rebate type. For example, ABC enterprises for May 2021 against Vol Rebate on Radius category, did a total quantity of 150 units and transaction amount of $80,000. This object is available in API version 65.0 and later.

RebatePartnerSpecialPrcTrm

Represents information about the special pricing term for a ship and debit program. This object is available in API version 58.0 and later.

RebatePayment

Tracks if the payment has been generated for this member for back end processing. This object is available in API version 51.0 and later.

RebatePayoutAdjustment

Rebate amount adjustment that needs to be given manually. An adjustment is an amount added or subtracted to the calculated rebate amount. This object is available in API version 51.0 and later.

RebateProgram

The rebate program your organization runs with a single account, all accounts, or specific list of accounts. This object is available in API version 51.0 and later.

RebateProgramMember

The member of a rebate program. By virtue of being a member, the partner or business account is eligible to get rebate payments. For example, ABC Enterprises and HVAC Corp are members of GoldStone Volume Rebate Program. This object is available in API version 51.0 and later.

RebateProgramMemberPayout

The payout calculated for a member for the period. For example, $1000 Jan'21 payout for ABC enterprises. This object is available in API version 51.0 and later.

RebateProgramPayoutPeriod

The period of the payout calculation. For example, 1st to 31st Jan, or 1st Dec to 14th Dec. This object is available in API version 51.0 and later.

RebatePtnrSpclPrcTrmBnft

Represents information about the benefit tier in a rebate partner special pricing term. This object is available in API version 58.0 and later.

ReceivedDocument

Allows partners to upload .CSV document. This object is available in API version 51.0 and later.

TransactionJournal

The transactions that need to be processed for a rebate program. For example, order line for 1000 units of $1200 for member ABC enterprises. This object is available in API version 51.0 and later.

UnitOfMeasureConversion

Represents the information used to convert a measurement value from a unit of measure to another. This object is available in API version 52.0 and later.

AccountAccountRelation

Represents a relationship between accounts, such as a relationship between a distributor account and a ship-to account. This object is available in API version 58.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
`AccountId`	**Type** reference **Properties** Create, Filter, Group, Sort, Update **Description** ID of the account associated with this account account relationship. This field is a relationship field. **Relationship Name** Account **Relationship Type** Lookup **Refers To** Account
`EndDate`	**Type** date **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** The date when the relationship ends.
`HierarchyType`	**Type** picklist **Properties** Create, Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort, Update **Description** Specifies the hierarchy between accounts that are related. Possible values are: • `Child`

Field	Details
	• `Parent` • `Peer` The default value is `Peer`.
`IsActive`	**Type** boolean **Properties** Create, Defaulted on create, Filter, Group, Sort, Update **Description** Indicates whether the account is actively involved with the related account (true) or not (false). The default value is `false`.
`LastReferencedDate`	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** The timestamp when the current user last accessed this record, a record related to this record, or a list view.
`LastViewedDate`	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** The timestamp when the current user last viewed this record or list view.
`Name`	**Type** string **Properties** Autonumber, Defaulted on create, Filter, idLookup, Sort **Description** Name of the account account relationship.
`OwnerId`	**Type** reference **Properties** Create, Defaulted on create, Filter, Group, Sort, Update **Description** ID of the owner of this object.

Field	Details
	This field is a polymorphic relationship field. **Relationship Name** Owner **Relationship Type** Lookup **Refers To** Group, User
`PartyRoleRelationId`	**Type** reference **Properties** Create, Filter, Group, Sort, Update **Description** The relationship between two accounts. This field is a relationship field. **Relationship Name** PartyRoleRelation **Relationship Type** Lookup **Refers To** PartyRoleRelation
`RelatedAccountId`	**Type** reference **Properties** Create, Filter, Group, Sort, Update **Description** The related account in the relationship. This field is a relationship field. **Relationship Name** RelatedAccount **Relationship Type** Lookup **Refers To** Account
`RelatedInverseRecordId`	**Type** reference **Properties** Create, Filter, Group, Nillable, Sort, Update

Field	Details
	Description The record that specifies the inverse relationship between the accounts. This field is a relationship field. **Relationship Name** RelatedInverseRecord **Relationship Type** Lookup **Refers To** AccountAccountRelation
`StartDate`	**Type** date **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** The date when the relationship starts.

PartyRoleRelation

Represents information about the type of relationship between the participants in a relationship. This object is available in API version 58.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
`LastReferencedDate`	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** The timestamp when the current user last accessed this record, a record related to this record, or a list view.

Field	Details
`LastViewedDate`	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** The timestamp when the current user last viewed this record or list view.
`Name`	**Type** string **Properties** Create, Filter, Group, idLookup, Sort, Update **Description** Name of the party role relationship.
`OwnerId`	**Type** reference **Properties** Create, Defaulted on create, Filter, Group, Sort, Update **Description** ID of the owner of this object. This field is a polymorphic relationship field. **Relationship Name** Owner **Relationship Type** Lookup **Refers To** Group, User
`RelatedInverseRecordId`	**Type** reference **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** The record that specifies the inverse relationship between the roles. This field is a relationship field. **Relationship Name** RelatedInverseRecord **Relationship Type** Lookup **Refers To** PartyRoleRelation

Field	Details
`RelatedRoleName`	**Type** string **Properties** Create, Filter, Group, Sort, Update **Description** The role that's related to another role in the relationship.
`RelationshipObjectName`	**Type** picklist **Properties** Create, Defaulted on create, Filter, Group, Restricted picklist, Sort, Update **Description** The object that's associated with the relationship. Possible values are: • `Account_Account_Relationship`—Account Account Relationship • `Contact_Contact_Relationship`—Contact Contact Relationship The default value is `Account_Account_Relationship`.
`RoleName`	**Type** string **Properties** Create, Filter, Group, Sort, Update **Description** The name of the role in the relationship.
`ShouldCreaInversRoleAuto`	**Type** boolean **Properties** Create, Defaulted on create, Filter, Group, Sort **Description** Indicates whether a role record should be created automatically for the relationship (true) or not (false). The default value is `false`.

ProgramRebateType

Provide the rebate types that are part of this program. For example, volume rebate, revenue rebate, or rebate on every transaction. This object is available in API version 51.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
`AccrualRate`	**Type** double **Properties** Create, Filter, Nillable, Sort, Update **Description** The rate you want to accrue at for the rebate program member. This rate is in the same units as the specified measure type, for example percent or amount. This field is available in API version 52.0 and later.
`AggregateObjectName`	**Type** picklist **Properties** Create, Filter, Group, Nillable, Restricted picklist, Sort, Update **Description** Name of the custom aggregation entity. For example, RebateMemberProductAggregate.
`BenefitQualifierField`	**Type** picklist **Properties** Create, Filter, Group, Nillable, Restricted picklist, Sort, Update **Description** Represents an attribute in the aggregate object whose value will be used, in conjunction with Aggregate Field and Rebate Type Benefit Field attributes from the PgmRebateTypBnftMapping entity, to determine the Benefit Value from the ProgramRebateTypeBenefit entity. Possible values are: • `GrowthAmount`—Rebate Member Product Aggregate:Growth Amount. • `GrowthAmountPercent`—Rebate Member Product Aggregate:Growth Amount Percent. • `GrowthQuantity`—Rebate Member Product Aggregate:Growth Quantity. • `GrowthQuantityPercent`—Rebate Member Product Aggregate:Growth Quantity Percent. • `PriorTotalQuantity`—Rebate Member Product Aggregate:Prior Total Quantity. • `PriorTotalTransactionAmount`—Rebate Member Product Aggregate:Prior Total Transaction Amount.

Field	Details
	• `TotalQuantity`—Rebate Member Product Aggregate:Total Quantity. • `TotalTransactionAmount`—Rebate Member Product Aggregate:Total Transaction Amount.
`CalcObjectId`	**Type** reference **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** Specifies the data processing engine definition used for the calculation. Only active data processing engine definitions can be used. Possible values are: • `Aggregate By Member` • `Aggregate By Product` • `Account Hierarchy Member Aggregate` • `Process Per Transaction` • `Ship and Debit Member Aggregate` • `Year on Year Growth`
`CalculationBasis`	**Type** picklist **Properties** Create, Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort, Update **Description** Specify the basis for rebate amount calculation. This field is available in API version 52.0 and later. Possible values are: • `Accrual`: Accrual rate, if provided, or the benefit setup is used to calculate the rebate amount. • `Payout`: Benefit setup is used to calculate the rebate amount. • `PayoutAndAccrual`—Payout and Accrual: Benefit setup is used for payout , and accrual rate for accrual amount. The default value is 'Payout'.
`CalculationMethod`	**Type** picklist **Properties** Create, Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort, Update **Description** Specifies the rule for calculating rebate benefits.

Field	Details
	Possible values are: • `Retrospective` • `Stepped`
`CalculationType`	**Type** picklist **Properties** Create, Filter, Group, Restricted picklist, Sort, Update **Description** Type of calculation for the rebate type. Possible values are: • `AggregateBased` • `Custom` • `GrowthBased` • `PerTransaction`
`FilterCriteria`	**Type** textarea **Properties** Nillable **Description** Stores the rebate type filter definition in JSON format, which will be consumed by the Calc object.
`FilterLogic`	**Type** string **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** Contains the filter criteria options that are offered - AND, OR, Custom.
`IsIntegratable`	**Type** boolean **Properties** Create, Defaulted on create, Filter, Group, Sort, Update **Description** This field determines if the rebate type record is of type template or not, based on which the record can be associated with an opportunity,contract, quote, or any other eligible object. The default value is 'false'.

Field	Details
`MeasureField`	**Type** picklist **Properties** Create, Filter, Group, Nillable, Restricted picklist, Sort, Update **Description** Stores the reference to the aggregate object that will be used for rebate amount calculations based on the measure type selected and the benefit value fetched. Possible values are: • `GrowthAmount`—Rebate Member Product Aggregate:Growth Amount. • `GrowthAmountPercent`—Rebate Member Product Aggregate:Growth Amount Percent. • `GrowthQuantity`—Rebate Member Product Aggregate:Growth Quantity. • `GrowthQuantityPercent`—Rebate Member Product Aggregate:Growth Quantity Percent. • `PriorTotalQuantity`—Rebate Member Product Aggregate:Prior Total Quantity. • `PriorTotalTransactionAmount`—Rebate Member Product Aggregate:Prior Total Transaction Amount. • `TotalQuantity`—Rebate Member Product Aggregate:Total Quantity. • `TotalTransactionAmount`—Rebate Member Product Aggregate:Total Transaction Amount.
`Name`	**Type** string **Properties** Create, Filter, Group, idLookup, Sort, Update **Description** Name of the program rebate type.
`ProductFilterType`	**Type** picklist **Properties** Create, Filter, Group, Nillable, Restricted picklist, Sort, Update **Description** The criteria to be applied for the products present in the Included or Excluded Products list. Possible values are: • `ExcludeProducts`—Exclude Products • `IncludeProducts`—Include Products
`RebateMeasureType`	**Type** picklist

Field	Details
	Properties Create, Filter, Group, Restricted picklist, Sort, Update **Description** Type of rebate. Possible values are: • `AmountperUnit` • `Custom` • `FixedAmount` • `PercentageOfRevenue`
`RebateProgramId`	**Type** reference **Properties** Create, Filter, Group, Sort, Update **Description** Reference to the rebate program.
`Status`	**Type** picklist **Properties** Create, Defaulted on create, Filter, Group, Restricted picklist, Sort, Update **Description** Status of the rebate type. Possible values are: • `Active` • `Inactive`
`UnitOfMeasureId`	**Type** reference **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** The unit of measure associated with the program rebate type. This field is available in API version 52.0 and later. This is a relationship field. **Relationship Name** UnitOfMeasure **Relationship Type** Lookup

Field	Details
	Refers To UnitOfMeasure

Usage

Each program can have multiple rebate types. For example, buy 100 units and get 2% revenue back, buy 10 units of petrol engines and get 1% rebate, and so on. If you want to give tiered benefits, select the Benefit Qualifier Field. For example, to give a tiered benefit of $5 for 0–100 units and $10 for 101–500 units, select Total Quantity in the Benefit Qualifier Field when the Measure Type is Total Quantity.

ProgramRebateTypeBenefit

Defines the benefit matrix for the rebate type. For example, 5% or $200. This object is available in API version 51.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
`BenefitValue`	**Type** double **Properties** Create, Filter, Sort, Update **Description** Rebate benefit at a level. For example, 3%, $3000.
`EffectiveEndDate`	**Type** date **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** The date until when the program rebate type benefit is effective. If the date does not match a payout period end date, it isn't picked.
`EffectiveStartDate`	**Type** date **Properties** Create, Filter, Group, Nillable, Sort, Update

Field	Details
	Description The date from when the program rebate type benefit is effective. If the date does not match a payout period start date, it isn't picked.
`MaximumMeasureFieldValue`	**Type** double **Properties** Create, Filter, Nillable, Sort, Update **Description** Cap the measure value at a certain figure. For example, pay 1% on $100,000 even when the members do more than that.
`MaximumQualifyingValue`	**Type** double **Properties** Create, Filter, Nillable, Sort, Update **Description** Upper limit of quantity or revenue at a level.
`MinimumQualifyingValue`	**Type** double **Properties** Create, Filter, Sort, Update **Description** Minimum quantity or revenue required to qualify to a level.
`Name`	**Type** string **Properties** Create, Filter, Group, idLookup, Sort, Update **Description** Name of the qualification. For example, Tier 1/2/3, Gold/Silver.
`ProductId`	**Type** reference **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** Use this if the benefits vary by products. For example, Product A has a tier from 1- 100 units for $20 and Product B has 1-100 units at $19. To use product or any dimension to the benefits, add that field in benefit mapping

Field	Details
`Status`	**Type** picklist **Properties** Create, Defaulted on create, Filter, Group, Restricted picklist, Sort, Update **Description** Status of the program rebate type benefit. Possible values are: • `Active` • `Inactive`

Usage

The benefit table can be extended with custom columns, for example, Region. The benefits can then be configured as: APAC 3%, USA 2%, and so on.

PgmRebateTypBnftMapping

If benefit table is extended, defines mapping of benefit field to the aggregate object fields. This object is available in API version 51.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
`AggregateField`	**Type** picklist **Properties** Create, Filter, Group, Restricted picklist, Sort, Update **Description** Used to describe the join conditions between the aggregate object and the Program Rebate Type Benefit entity. Possible values are: • `RebateMemberProductAggregate.GrowthAmount`—Growth Amount • `RebateMemberProductAggregate.GrowthAmountPercent`—Growth Amount Percent

Field	Details
	• `RebateMemberProductAggregate.GrowthQuantity`—Growth Quantity • `RebateMemberProductAggregate.GrowthQuantityPercent`—Growth Quantity Percent • `RebateMemberProductAggregate.LastReferencedDate`—Last Referenced Date • `RebateMemberProductAggregate.LastViewedDate`—Last Viewed Date • `RebateMemberProductAggregate.PriorTotalQuantity`—Prior Total Quantity • `RebateMemberProductAggregate.PriorTotalTransactionAmount`—Prior Total Transaction Amount • `RebateMemberProductAggregate.ProductId`—Product ID • `RebateMemberProductAggregate.ProgramRebateTypeId`—Program Rebate ID • `RebateMemberProductAggregate.RebateProgramMemberId`—Rebate Program Member ID • `RebateMemberProductAggregate.RebateProgramPayoutPeriodId`—Rebate Program Payout Period ID • `RebateMemberProductAggregate.TotalQuantity`—Total Quantity • `RebateMemberProductAggregate.TotalTransactionAmount`—Total Transaction Amount • `RebateMemberProductAggregate.TransactionJournalId`— ID
`ProgramRebateTypeId`	**Type** reference **Properties** Create, Filter, Group, Sort **Description** Parent identifier
`RebateTypeBenefitField`	**Type** picklist **Properties** Create, Filter, Group, Restricted picklist, Sort, Update **Description** Used to describe the join conditions between the aggregate object and the Program Rebate Type Benefit entity. Possible values are: • `ProductId`—ID of the Product

ProgramRebateTypeFilter

The definition that filters the transaction journals eligible for a rebate type. This filter definition is used in the rebates data processing engines. This object is available in API version 51.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
`FilterField`	**Type** picklist **Properties** Create, Filter, Group, Nillable, Restricted picklist, Sort, Update **Description** The field name to use for the program rebate type filter. Possible values are: • `ActivityDate`—Activity date. • `CloneSourceId`—Cloned source ID. • `CreatedById`—Created by. • `CreatedDate`—Created date. • `ExternalTransactionNumber`—External transaction ID. • `Id`— ID • `IsDeleted`—Deleted. • `JournalDate`—Journal date. • `LastModifiedById`—Last modified by. • `LastModifiedDate`—The most recent date on which a user modified this record. • `LastReferencedDate`—The most recent date on which a user referenced this record. • `LastViewedDate`—The most recent date on which a user viewed this record. • `MemberId`—Program Member • `Name` • `OrderId`—Order • `OrderItemId`—Order Product. • `ProductCategoryId`—Product Category • `ProductId`—Product • `Quantity`

Field	Details
	• `QuantityUnitOfMeasureId`—Quantity Unit • `SystemModstamp`—System Modstamp • `TransactionAmount`—Transaction Amount. • `UsageType`—Usage Type. • `UserRecordAccessId`—Object Access Level.
`Operator`	**Type** picklist **Properties** Create, Filter, Group, Nillable, Restricted picklist, Sort, Update **Description** The filter operator. Possible values are: • `Contains`—in • `Equals`—equals • `GreaterThan`—greater than • `GreaterThanOrEquals`—greater or equal • `LessThan`—less than • `LessThanOrEquals`—less or equal • `NotEquals`—not equal to
`ProgramRebateTypeId`	**Type** reference **Properties** Create, Filter, Group, Sort **Description** The parent program rebate type associated with the filter.
`Sequence`	**Type** int **Properties** Create, Filter, Group, Sort, Update **Description** The unique sequence number for each condition within the program rebate type filter.
`Value`	**Type** string **Properties** Create, Filter, Group, Sort, Update

Field	Details
	Description The value for filtering the program rebate type.

ProgramRebateTypePayout

The payout given to a member for a particular rebate type. For example, volume rebate payout in Jan'19 for ABC enterprises is $560, petrol engine payout for ABC in Jan'19 is $440. This object is available in API version 51.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
`AccruedAmount`	**Type** currency **Properties** Filter, Nillable, Sort **Description** The accrued amount for the program rebate type payout. This is a calculated field.
`Name`	**Type** string **Properties** Create, Filter, Group, idLookup, Sort, Update **Description** Name of the program rebate type.
`ProgramRebateTypeId`	**Type** reference **Properties** Create, Filter, Group, Sort **Description** Unique identifier for the program rebate type.

Field	Details
RebateAmount	**Type** currency **Properties** Filter, Nillable, Sort **Description** Rebate amount. This is a calculated field.
RebateProgramMemberPayoutId	**Type** reference **Properties** Create, Filter, Group, Sort **Description** Unique identifier of the rebate program member payout.
TotalBenefitQualifierValue	**Type** double **Properties** Filter, Nillable, Sort **Description** Roll-up summary of the Benefit Qualifier Value from the Program Rebate Type Payout Source entity. This is a calculated field.
TotalMeasureValue	**Type** double **Properties** Filter, Nillable, Sort **Description** Roll-up summary for the Measure Value field from the Program Rebate Type Payout Source entity. This is a calculated field.

ProgramRebateTypPayoutSrc

The rebate amount and the tier applied calculated for each row in the aggregate. There is a 1 to 1 relation between payout source and aggregate. This object is available in API version 51.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
`AccrualRate`	**Type** double **Properties** Create, Filter, Nillable, Sort, Update **Description** The accrual rate for the program rebate type payout source. This field is available in API version 52.0 and later.
`AccruedAmount`	**Type** currency **Properties** Create, Filter, Nillable, Sort, Update **Description** The accrued amount for the program rebate type payout source. This field is available in API version 52.0 and later.
`AggregateIdentifierId`	**Type** reference **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** Unique identifier for the aggregate row.
`BenefitQualifierValue`	**Type** double **Properties** Create, Filter, Nillable, Sort, Update **Description** Value of the attribute from the aggregate object that is set up as Benefit Qualifier.
`MeasureValue`	**Type** double **Properties** Create, Filter, Nillable, Sort, Update

Field	Details
	Description Value of the attribute from the aggregate object that is set up as Measure Field.
`MemberId`	**Type** reference **Properties** Create, Filter, Group, Sort **Description** Unique identifier for the member.
`Name`	**Type** string **Properties** Create, Filter, Group, idLookup, Sort, Update **Description** Name of the rebate program member.
`ProgramRebateTypeBenefitId`	**Type** reference **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** Program rebate type benefit identifier.
`ProgramRebateTypePayoutId`	**Type** reference **Properties** Create, Filter, Group, Sort **Description** Unique identifier of the program rebate type payout.
`RebateAmount`	**Type** currency **Properties** Create, Filter, Nillable, Sort, Update **Description** Rebate amount calculated by invocable actions.

ProgramRebateTypeProduct

Represents a junction between a program rebate type and a product. This object is available in API version 53.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
`ProductId`	**Type** reference **Properties** Create, Filter, Group, Sort, Update **Description** The product associated with the record. This is a relationship field. **Relationship Name** Product **Relationship Type** Lookup **Refers To** Product2
`ProgramRebateTypeId`	**Type** reference **Properties** Create, Filter, Group, Sort **Description** The program rebate type associated with the product. This is a relationship field. **Relationship Name** ProgramRebateType **Relationship Type** Lookup **Refers To** ProgramRebateType

ProgramRebateTypeReference

Represents the association between the contract, opportunity, or any eligible standard or custom object, and rebate type. This object is available in API version 52.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
`ProgramRebateTypeId`	**Type** reference **Properties** Create, Filter, Group, Sort, Update **Description** The program rebate type associated with the record. This is a relationship field. **Relationship Name** ProgramRebateType **Relationship Type** Lookup **Refers To** ProgramRebateType
`ProgramRebateTypeSrcId`	**Type** reference **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** The source program rebate type associated to the contract, opportunity, or quote, or any other eligible standard or custom object. This is a relationship field. **Relationship Name** ProgramRebateTypeSrc **Relationship Type** Lookup **Refers To** ProgramRebateType

Field	Details
`ReferenceObjectId`	**Type** reference **Properties** Create, Filter, Group, Sort, Update **Description** The contract, opportunity, quote, or any other eligible standard or custom object associated with the program rebate type. This is a polymorphic relationship field. **Relationship Name** ReferenceObject **Relationship Type** Lookup **Refers To** Contract, Opportunity, Order, SalesAgreement

RebateClaim

Represents information about the claim submitted by the end customer or distributor for a ship and debit program. This object is available in API version 54.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
`ApprovedAmount`	**Type** currency **Properties** Filter, Nillable, Sort **Description** The approved rebate claim amount. Available in API version 58.0 and later.
`CalculatedAmount`	**Type** currency **Properties** Create, Filter, Nillable, Sort, Update

Field	Details
	Description The calculated rebate claim amount. Available in API version 58.0 and later.
`ClaimAmount`	**Type** currency **Properties** Create, Filter, Sort, Update **Description** The amount for which rebate is claimed.
`ClaimDate`	**Type** dateTime **Properties** Create, Filter, Sort, Update **Description** The date when the claim is submitted by the claimant.
`ClaimType`	**Type** picklist **Properties** Create, Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort, Update **Description** Specifies the type of the rebate claim. Available in API version 58.0 and later. Possible values are: • `ShipAndDebit`—Ship and Debit • `StandardRebate`—Standard Rebate The default value is `StandardRebate`.
`ClaimedByAccountId`	**Type** reference **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** The account of the customer or distributor who is submitting the claim. This field is a relationship field. **Relationship Name** ClaimedByAccount **Relationship Type** Lookup

Field	Details
	Refers To Account
`ClaimedByAccountNumber`	**Type** string **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** The reference number of the account of the customer or distributor who is submitting the claim.
`CurrencyIsoCode`	**Type** picklist **Properties** Create, Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort, Update **Description** If multiple currencies are enabled, this field contains the currency ISO code associated with the record. Possible values are: • `EUR`—Euro • `USD`—U.S. Dollar The default value is `EUR`.
`Description`	**Type** textarea **Properties** Create, Filter, Nillable, Sort, Update **Description** The description of the rebate claim.
`LastReferencedDate`	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** The most recent date on which a user referenced this record.
`LastViewedDate`	**Type** dateTime **Properties** Filter, Nillable, Sort

Field	Details
	Description The most recent date on which a user viewed this record.
`Name`	**Type** string **Properties** Autonumber, Defaulted on create, Filter, idLookup, Sort **Description** Autogenerated name of the rebate claim.
`OwnerId`	**Type** reference **Properties** Create, Defaulted on create, Filter, Group, Sort, Update **Description** ID of the rebate claim record owner. This field is a polymorphic relationship field. **Relationship Name** Owner **Relationship Type** Lookup **Refers To** Group, User
`ParticipatingAccountId`	**Type** reference **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** The account of a participant in the claim. It can be the ship to/end customer or shipper/channel partner. This field is a relationship field. **Relationship Name** ParticipatingAccount **Relationship Type** Lookup **Refers To** Account
`ParticipatingAccountNumber`	**Type** string

Field	Details
	Properties Create, Filter, Group, Nillable, Sort, Update **Description** The reference number of the account of a participant in the claim. It can be the ship to/end customer or shipper/ channel partner.
`ProcessingStatus`	**Type** picklist **Properties** Create, Defaulted on create, Filter, Group, Nillable, Sort, Update **Description** The processing status of the rebate claim. Possible values are: • `Approved` • `Complete` • `In Review` • `New` • `Rejected` • `Reviewed` • `System Failed` • `System Processed` The default value is `New`.
`ProcessingStatusMessage`	**Type** string **Properties** Create, Filter, Nillable, Sort, Update **Description** The message with the details of the rebate claim's processing status when it isn't completed.
`ProductCode`	**Type** string **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** The reference number of the product for which the rebate claim is submitted.
`ProductId`	**Type** reference

Field	Details
	Properties Create, Filter, Group, Nillable, Sort, Update **Description** The product for which the rebate claim is submitted. This field is a relationship field. **Relationship Name** Product **Relationship Type** Lookup **Refers To** Product2
`ProgramReferenceNumber`	**Type** string **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** The reference number of the rebate program.
`Quantity`	**Type** double **Properties** Create, Filter, Sort, Update **Description** The quantity of the product for which the rebate claim is submitted.
`QuantityUnitOfMeasureId`	**Type** reference **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** The unit of measure of the quantity. Available in API version 58.0 and later. **Relationship Name** QuantityUnitOfMeasure **Relationship Type** Lookup **Refers To** UnitOfMeasure
`RebateMemberClaimAggregateId`	**Type** reference

Field	Details
	Properties Create, Filter, Group, Nillable, Sort, Update **Description** The rebate member claim aggregate record related to the rebate claim. Available in API version 58.0 and later. **Relationship Name** RebateMemberClaimAggregate **Relationship Type** Lookup **Refers To** RebateMemberClaimAggregate
`RebatePaymentId`	**Type** reference **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** The rebate payment record related to the rebate claim. Available in API version 58.0 and later. **Relationship Name** RebatePayment **Relationship Type** Lookup **Refers To** RebatePayment
`ReceivedDocumentId`	**Type** reference **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** The received document that this claim is part of. Available in API version 58.0 and later. **Relationship Name** ReceivedDocument **Relationship Type** Lookup **Refers To** ReceivedDocument
`ReferencePricePerUnit`	**Type** currency

Field	Details
	Properties Create, Filter, Nillable, Sort, Update **Description** The price of a product unit that's agreed upon between the manufacturer and the claimant by the manufacturer and the distributor or the manufacturer and the customer for the rebate claim.
`ResalePricePerUnit`	**Type** currency **Properties** Create, Filter, Nillable, Sort, Update **Description** The price at which the claimant bought the product.
`SalePricePerUnit`	**Type** currency **Properties** Create, Filter, Nillable, Sort, Update **Description** The original price at which a product unit is sold to the customer by the manufacturer.
`ShipDate`	**Type** dateTime **Properties** Create, Filter, Nillable, Sort, Update **Description** The date when the product was shipped to end customer. Available in API version 58.0 and later.
`ShippedTransactionIdentifier`	**Type** string **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** The identifier of the transaction between distributor and end customer or ship to account. Available in API version 58.0 and later.
`TotalAdjustmentAmount`	**Type** currency **Properties** Create, Filter, Nillable, Sort, Update

Field	Details
	Description The total amount adjusted on the rebate claim. Available in API version 58.0 and later.
`TransactionIdentifier`	**Type** string **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** The unique identifier of the transaction between the manufacturer and distributor. Available in API version 58.0 and later.
`UsageType`	**Type** picklist **Properties** Create, Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort **Description** Specifies the usage type of the rebate claim. Possible value is: • `Rebates` The default value is `Rebates`. Available in API version 58.0 and later.

RebateClaimAdjustment

Represents information about the adjustments made to the rebate claim. This object is available in API version 58.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
`AdjustmentAmount`	**Type** currency **Properties** Create, Filter, Sort, Update **Description** The adjustment amount for the rebate claim.

Field	Details
`ApprovedDate`	**Type** date **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** The date when the rebate claim adjustment was approved.
`Comments`	**Type** string **Properties** Create, Filter, Group, Sort, Update **Description** The details about the adjustment made to the rebate claim.
`CurrencyIsoCode`	**Type** picklist **Properties** Create, Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort, Update **Description** The currency ISO code associated with the record. This is available when multiple currencies are enabled. Possible values are: • `EUR`—Euro • `USD`—U.S. Dollar The default value is `EUR`.
`LastReferencedDate`	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** The most recent date on which a user referenced this record.
`LastViewedDate`	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** The most recent date on which a user viewed this record.

Field	Details
`Name`	**Type** string **Properties** Create, Filter, Group, idLookup, Sort, Update **Description** The name of the rebate claim adjustment.
`RebateClaimId`	**Type** reference **Properties** Create, Filter, Group, Sort **Description** The rebate claim record to which the adjustment was made. This field is a relationship field. **Relationship Name** RebateClaim **Relationship Type** Lookup **Refers To** RebateClaim
`Status`	**Type** picklist **Properties** Create, Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort, Update **Description** The status of the adjustment made to the rebate claim. Possible values are: • `Approved` • `InReview`—Under Review • `Rejected` The default value is `InReview`.

RebateMemberAggregateItem

Represents a junction between a rebate member product aggregate and a transaction journal. This object is available in API version 54.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
`LastReferencedDate`	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** The timestamp when the current user last accessed this record indirectly, for example, through a list view or related record.
`LastViewedDate`	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** The timestamp when the current user last viewed this record or list view. If this value is null, and `LastReferenceDate` is not null, the user accessed this record or list view indirectly.
`Name`	**Type** string **Properties** Autonumber, Defaulted on create, Filter, idLookup, Sort **Description** Name of the rebate member aggregate item.
`OwnerId`	**Type** reference **Properties** Create, Defaulted on create, Filter, Group, Sort, Update **Description** ID of the owner of this object. This field is a polymorphic relationship field. **Relationship Name** Owner **Relationship Type** Lookup

Field	Details
	Refers To Group, User
RebateAmount	**Type** currency **Properties** Create, Filter, Nillable, Sort, Update **Description** The rebate amount for the associated transaction journal record.
RebateMemberProductAggregateId	**Type** reference **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** The rebate member product aggregate record associated with the transaction journal record. This is a relationship field. **Relationship Name** RebateMemberProductAggregate **Relationship Type** Lookup **Refers To** RebateMemberProductAggregate
TransactionAmount	**Type** currency **Properties** Create, Filter, Nillable, Sort, Update **Description** The transaction amount for the transaction journal record associated with the aggregate item record.
TransactionJournalId	**Type** reference **Properties** Create, Filter, Group, Sort, Update **Description** The transaction journal associated with the rebate aggregated transaction journal record. This is a relationship field. **Relationship Name** TransactionJournal

Field	Details
	Relationship Type Lookup **Refers To** TransactionJournal
`TransactionQuantity`	**Type** double **Properties** Create, Filter, Nillable, Sort, Update **Description** The transaction quantity of the associated transaction journal record.

RebateMemberClaimAggregate

Represents information about the aggregated claim quantity and amount for a rebate member per product. This object is available in API version 58.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
`CurrencyIsoCode`	**Type** picklist **Properties** Create, Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort, Update **Description** The currency ISO code associated with the record. This is available when multiple currencies are enabled. Possible values are: • `EUR`—Euro • `USD`—U.S. Dollar The default value is `EUR`.
`LastReferencedDate`	**Type** dateTime

Field	Details
	Properties Filter, Nillable, Sort **Description** The most recent date on which a user referenced this record.
`LastViewedDate`	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** The most recent date on which a user viewed this record.
`Name`	**Type** string **Properties** Autonumber, Defaulted on create, Filter, idLookup, Sort **Description** The auto-generated name of the rebate member claim aggregate.
`OwnerId`	**Type** reference **Properties** Create, Defaulted on create, Filter, Group, Sort, Update **Description** The ID of the rebate member claim aggregate record owner. This field is a polymorphic relationship field. **Relationship Name** Owner **Relationship Type** Lookup **Refers To** Group, User
`PriceType`	**Type** Picklist **Properties** Create, Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort, Update. **Description** Specifies the basis on which the product was sold in a rebate claim. Used to distinguish between standard and contract pricing strategies when calculating rebate eligibility.

Field	Details
`ProductId`	**Type** reference **Properties** Create, Filter, Group, Sort, Update **Description** The product for which the rebate claim was made. This field is a relationship field. **Relationship Name** Product **Relationship Type** Lookup **Refers To** Product2
`RebateProgramMemberId`	**Type** reference **Properties** Create, Filter, Group, Sort, Update **Description** The rebate program member for whom the claim was made. This field is a relationship field. **Relationship Name** RebateProgramMember **Relationship Type** Lookup **Refers To** RebateProgramMember
`TotalClaimedAmount`	**Type** currency **Properties** Create, Filter, Sort, Update **Description** The total amount that's already claimed.
`TotalClaimedQuantity`	**Type** double **Properties** Create, Filter, Sort, Update **Description** The total product quantity that's already claimed.

RebateMemberProductAggregate

Stores the post calculation summary of journal transactions by member, period, and rebate type. For example, ABC enterprises for May 2021 against Vol Rebate on Radius category, did a total quantity of 150 units and transaction amount of $80,000. This object is available in API version 65.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
`AggregateReferenceNumber`	**Type** string **Properties** Create, Filter, Group, idLookup, Nillable, Sort, Update **Description** A unique reference number of the aggregate record that's assigned by the Data Processing Engine when generating Rebate Program Member Aggregate records. This reference number can be used to relate Rebate Member Aggregate Item records to the aggregate record. Available in API version 54.0 and later.
`GrowthAmount`	**Type** currency **Properties** Create, Filter, Nillable, Sort, Update **Description** The difference between the total amount and the prior amount in the rebate aggregate.
`GrowthAmountPercent`	**Type** double **Properties** Create, Filter, Nillable, Sort, Update **Description** The difference between the total amount and the prior amount in percentage in the rebate aggregate.
`GrowthQuantity`	**Type** double **Properties** Create, Filter, Nillable, Sort, Update

Field	Details
	Description The difference between the total quantity and the previous quantity in the transaction journal associated with the rebate aggregate.
`GrowthQuantityPercent`	**Type** double **Properties** Create, Filter, Nillable, Sort, Update **Description** The quantity growth in percentage.
`LastReferencedDate`	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** The timestamp when the current user last accessed this record indirectly, for example, through a list view or related record.
`LastViewedDate`	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** The timestamp when the current user last viewed this record or list view. If this value is null, and `LastReferenceDate` is not null, the user accessed this record or list view indirectly.
`Name`	**Type** string **Properties** Autonumber, Defaulted on create, Filter, idLookup, Sort **Description** Name of the rebate member product aggregate.
`OwnerId`	**Type** reference **Properties** Create, Defaulted on create, Filter, Group, Sort, Update **Description** ID of the owner of this object. This field is a polymorphic relationship field.

Field	Details
	Relationship Name Owner **Relationship Type** Lookup **Refers To** Group, User
PriorTotalQuantity	**Type** double **Properties** Create, Filter, Nillable, Sort, Update **Description** The quantity in the corresponding payout period of the previous rebate cycle.
PriorTotalTransactionAmount	**Type** currency **Properties** Create, Filter, Nillable, Sort, Update **Description** The total transaction amount corresponding to the prior period.
ProductId	**Type** reference **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** The product associated with the rebate aggregation.
ProgramRebateTypeId	**Type** reference **Properties** Create, Filter, Group, Sort, Update **Description** The program rebate type associated with the rebate aggregation.
RebatePayoutStatus	**Type** picklist **Properties** Create, Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort, Update **Description** The status of the aggregate rebate payout to the member.

Field	Details
	Possible values are: • `Error` • `InProgress` • `Success` The default value is `InProgress`.
`RebatePayoutStatusMessage`	**Type** string **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** The status message of the aggregate rebate payout to the member.
`RebateProgramMemberId`	**Type** reference **Properties** Create, Filter, Group, Sort, Update **Description** The rebate program member associated with the rebate aggregation.
`RebateProgramPayoutPeriodId`	**Type** reference **Properties** Create, Filter, Group, Sort, Update **Description** The rebate program payout period associated with the rebate aggregation.
`RollupProgramMemberId`	**Type** reference **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** The immediate parent account in the hierarchy of the account (rebate program member) associated with the product aggregate. This field is available in API version 52.0 and later. This is a relationship field. **Relationship Name** RollupProgramMember **Relationship Type** Lookup **Refers To** RebateProgramMember

Field	Details
`TotalQuantity`	**Type** double **Properties** Create, Filter, Nillable, Sort, Update **Description** The sum of values in transaction journal quantity fields associated with the rebate aggregation.
`TotalTransactionAmount`	**Type** currency **Properties** Create, Filter, Nillable, Sort, Update **Description** The sum of values in transaction journal transaction amount fields associated with the rebate aggregation.
`TransactionJournalId`	**Type** reference **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** The transaction journal associated with the program rebate type.

RebatePartnerSpecialPrcTrm

Represents information about the special pricing term for a ship and debit program. This object is available in API version 58.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
`CurrencyIsoCode`	**Type** picklist **Properties** Create, Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort, Update

Field	Details
	Description The currency ISO code associated with the record. This is available when multiple currencies are enabled. Possible values are: • `EUR`—Euro • `USD`—U.S. Dollar The default value is `EUR`.
`Discount`	**Type** double **Properties** Create, Filter, Nillable, Sort, Update **Description** The agreed discount for a product unit between the manufacturer and the distributor as part of the special pricing terms for a ship and debit program. Between `ReferencePricePerUnit` and `Discount`, only one field can be added or used.
`IsDiscountByPercent`	**Type** boolean **Properties** Create, Defaulted on create, Filter, Group, Sort, Update **Description** Indicates whether the discount is given by percentage (`true`) or amount (`false`). The default value is `false`.
`LastReferencedDate`	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** The most recent date on which a user referenced this record.
`LastViewedDate`	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** The most recent date on which a user viewed this record.
`Name`	**Type** string

Field	Details
	Properties Create, Filter, Group, idLookup, Sort, Update **Description** The user-defined name of the rebate partner special pricing term.
`OwnerId`	**Type** reference **Properties** Create, Defaulted on create, Filter, Group, Sort, Update **Description** The ID of the rebate partner special pricing term. This field is a polymorphic relationship field. **Relationship Name** Owner **Relationship Type** Lookup **Refers To** Group, User
`ProductId`	**Type** reference **Properties** Create, Filter, Group, Sort, Update **Description** The product associated with the rebate partner special pricing term. This field is a relationship field. **Relationship Name** Product **Relationship Type** Lookup **Refers To** Product2
`ProductQtyUnitOfMeasureId`	**Type** reference **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** The unit of measure associated with the product quantity. This field is a relationship field.

Field	Details
	Relationship Name ProductQtyUnitOfMeasure **Relationship Type** Lookup **Refers To** UnitOfMeasure
`ProductQuantity`	**Type** double **Properties** Create, Filter, Sort, Update **Description** The threshold quantity for a product agreed between the manufacturer and the distributor as part of the special pricing terms for a ship and debit program.
`RebateProgramMemberId`	**Type** reference **Properties** Create, Filter, Group, Sort, Update **Description** The rebate program member associated with the rebate partner special pricing term. This field is a relationship field. **Relationship Name** RebateProgramMember **Relationship Type** Lookup **Refers To** RebateProgramMember
`ReferencePricePerUnit`	**Type** currency **Properties** Create, Filter, Nillable, Sort, Update **Description** The price of a product unit that's agreed upon between the manufacturer and the distributor as part of the special pricing term. Between ReferencePricePerUnit and Discount, only one field can be added or used.
`SalePricePerUnit`	**Type** currency

Field	Details
	Properties Create, Filter, Sort, Update **Description** The actual price of each unit of the product that's sold to the partner by the manufacturer.
`Status`	**Type** picklist **Properties** Create, Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort, Update **Description** The status of the rebate partner special pricing term. Possible values are: • `Active` • `Draft` • `Inactive` The default value is `Draft`.

RebatePayment

Tracks if the payment has been generated for this member for back end processing. This object is available in API version 51.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
`Amount`	**Type** currency **Properties** Create, Filter, Sort, Update **Description** Amount to be paid out.
`Name`	**Type** string

Field	Details
	Properties Create, Filter, Group, idLookup, Sort, Update **Description** Name of the rebate program.
`PaymentDate`	**Type** date **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** Date when payment is issued.
`Status`	**Type** picklist **Properties** Create, Filter, Group, Nillable, Restricted picklist, Sort, Update **Description** Status of the payout. Possible values are: • `Generated` • `Paid` • `Rejected`

RebatePayoutAdjustment

Rebate amount adjustment that needs to be given manually. An adjustment is an amount added or subtracted to the calculated rebate amount. This object is available in API version 51.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
`AdjustmentAmount`	**Type** currency

Field	Details
	Properties Create, Filter, Sort, Update **Description** Value of the adjustment.
`ApprovedDate`	**Type** date **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** Approved date of the adjustment.
`Comments`	**Type** string **Properties** Create, Filter, Group, Sort, Update **Description** Notes for adjustment.
`Name`	**Type** string **Properties** Create, Filter, Group, idLookup, Sort, Update **Description** Name of the rebate program.
`RebateProgramMemberPayoutId`	**Type** reference **Properties** Create, Filter, Group, Sort **Description** Rebate program member payout identifier.
`Status`	**Type** picklist **Properties** Create, Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort, Update **Description** Status of adjustment. Possible values are: • `Approved`

Field	Details
	• `InReview`—Under Review • `Rejected`

RebateProgram

The rebate program your organization runs with a single account, all accounts, or specific list of accounts. This object is available in API version 51.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
`Description`	**Type** textarea **Properties** Create, Filter, Nillable, Sort, Update **Description** The description of the rebate program.
`EndDate`	**Type** date **Properties** Create, Filter, Group, Sort, Update **Description** The end date of the rebate program.
`Frequency`	**Type** picklist **Properties** Create, Filter, Group, Restricted picklist, Sort, Update **Description** The frequency of rebate program. Possible values are: • `Annually` • `Biannually`

Field	Details
	• `CustomPeriods` • `Monthly` • `OnDemand` • `ProgramStartAndEndDate` • `Quarterly`
`GracePeriodDayCount`	**Type** int **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** Number of days from end date of an instance of a payout period. Transactions submitted until this day are considered for rebate calculation. For example, a value of 5 implies that channel partners can submit transactions for a monthly period before the 5th of the next month. This isn't enforced in the system but can be used as information for channel partners.
`Name`	**Type** string **Properties** Create, Filter, Group, idLookup, Sort, Update **Description** Name of the rebate program.
`PayoutCalculationDayCount`	**Type** int **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** Number of days from period end date. The payment will be calculated after this date. For example, a payout calculation day count of 14 means 14 days after period end date.
`ProgramReferenceNumber`	**Type** string **Properties** Create, Filter, Group, idLookup, Nillable, Sort, Update **Description** The reference number assigned to this rebate program. This field is unique within your organization. This field is available in API version 52.0 and later.
`RebateProgramUrl`	**Type** url

Field	Details
	Properties Create, Filter, Group, Nillable, Sort, Update **Description** URL of the rebate program.
`StartDate`	**Type** date **Properties** Create, Filter, Group, Sort, Update **Description** Start date of rebate program.
`Status`	**Type** picklist **Properties** Create, Defaulted on create, Filter, Group, Restricted picklist, Sort, Update **Description** Possible states of rebate program. Possible values are: • `Active` • `Draft` • `Inactive`
`Type`	**Type** picklist **Properties** Create, Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort, Update **Description** The type of the rebate program. Available in API version 58.0 and later. Possible values are: • `ShipAndDebit` • `Standard` The default value is `Standard`.

RebateProgramMember

The member of a rebate program. By virtue of being a member, the partner or business account is eligible to get rebate payments. For example, ABC Enterprises and HVAC Corp are members of GoldStone Volume Rebate Program. This object is available in API version 51.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
`AccountId`	**Type** reference **Properties** Create, Filter, Group, Sort, Update **Description** Unique identifier for the account who's a member of the program.
`CumulativePayoutAmount`	**Type** currency **Properties** Filter, Nillable, Sort **Description** Total rebate amount for a member. Sum of Total Payout Amount values from RebateProgramMemberPayout object that are not rejected. This is a calculated field.
`IsPayoutCalcSkipped`	**Type** boolean **Properties** **Description** Specifies whether the rebate payout calculation should be skipped for the rebate program member. This field is available in API version 52.0 and later.
`LastActivationDate`	**Type** date **Properties** Filter, Group, Nillable, Sort **Description** The latest date on which the program member was activated.
`LastDeactivationDate`	**Type** date **Properties** Filter, Group, Nillable, Sort

Field	Details
	Description The latest date on which the program member was deactivated.
`MemberStatus`	**Type** picklist **Properties** Create, Defaulted on create, Filter, Group, Restricted picklist, Sort, Update **Description** Status of membership. Possible values are: • `Active` • `Inactive`
`Name`	**Type** string **Properties** Create, Filter, Group, idLookup, Sort, Update **Description** Name of the member.
`RebateProgramId`	**Type** reference **Properties** Create, Filter, Group, Sort, Update **Description** Reference to rebate program identifier.
`ShipToAccountId`	**Type** reference **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** The ship to account. Use this when the rebate is applicable only when the member ships to this account. This field is available in API version 52.0 and later. This is a relationship field. **Relationship Name** ShipToAccount **Relationship Type** Lookup **Refers To** Account

RebateProgramMemberPayout

The payout calculated for a member for the period. For example, $1000 Jan'21 payout for ABC enterprises. This object is available in API version 51.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
`AccruedAmount`	**Type** currency **Properties** Filter, Nillable, Sort **Description** The payout amount accrued by the rebate program member. This is a calculated field.
`CalculatedRebateAmount`	**Type** currency **Properties** Filter, Nillable, Sort **Description** Total amount calculated. This is a calculated field.
`CalculationDate`	**Type** date **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** Date on which the calculation happened.
`MemberId`	**Type** reference **Properties** Create, Filter, Group, Sort **Description** Program member ID.

Field	Details
`Name`	**Type** string **Properties** Create, Filter, Group, idLookup, Sort, Update **Description** Name of the rebate program.
`Status`	**Type** picklist **Properties** Create, Defaulted on create, Filter, Group, Restricted picklist, Sort, Update **Description** Status of payout. Possible values are: • `Approved` • `Interim`—Calculation In progress. • `Pending`—Under review. • `Rejected` • `SystemCalculated`—Calculated.
`TotalApprovedAdjustmentAmount`	**Type** currency **Properties** Filter, Nillable, Sort **Description** Summarized payout adjustment amount. This is a calculated field.
`TotalRebateAmount`	**Type** currency **Properties** Filter, Nillable, Sort **Description** Summarized payout amount. This is a calculated field.

RebateProgramPayoutPeriod

The period of the payout calculation. For example, 1st to 31st Jan, or 1st Dec to 14th Dec. This object is available in API version 51.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
`EndDate`	**Type** date **Properties** Create, Filter, Group, Sort, Update **Description** End date of the period. The system picks transactions between start date and end date of the period for rebate calculations.
`LastCalculationDate`	**Type** date **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** Last date on which the rebate amount calculation for that period for all the relevant aggregate records was completed successfully.
`MemberPayoutCount`	**Type** int **Properties** Filter, Group, Nillable, Sort **Description** Used to track total number of payouts that happened in a period for a particular rebate program.
`PayoutCalculationDate`	**Type** date **Properties** Create, Filter, Group, Sort, Update **Description** Date on or after which the payout calculation for this period is set to Closed.

Field	Details
`RebateProgramId`	**Type** reference **Properties** Create, Filter, Group, Sort **Description** Rebate program identifier.
`StartDate`	**Type** date **Properties** Create, Filter, Group, Sort, Update **Description** Start date of the period.
`Status`	**Type** picklist **Properties** Create, Defaulted on create, Filter, Group, Restricted picklist, Sort, Update **Description** Used to track if a payout period is active or inactive. Possible values are: • `Active` • `Closed` • `Inactive`
`TotalApprovedAmount`	**Type** currency **Properties** Filter, Nillable, Sort **Description** Total approved member payout amount.
`TotalRejectedAmount`	**Type** currency **Properties** Filter, Nillable, Sort **Description** Total rejected amount.
`TransactionGracePeriodDate`	**Type** date

Field	Details
	Properties Create, Filter, Group, Sort, Update **Description** Cut-off date before which the transactions are accepted for the period.

RebatePtnrSpclPrcTrmBnft

Represents information about the benefit tier in a rebate partner special pricing term. This object is available in API version 58.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
`CurrencyIsoCode`	**Type** picklist **Properties** Create, Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort, Update **Description** The currency ISO code associated with the record. This is available when multiple currencies are enabled. Possible values are: • `EUR`—Euro • `USD`—U.S. Dollar The default value is `EUR`.
`Discount`	**Type** double **Properties** Create, Filter, Nillable, Sort, Update **Description** The discount that's applicable for the rebate partner special pricing term benefit.
`EffectiveEndDate`	**Type** dateTime

Field	Details
	Properties Create, Filter, Nillable, Sort, Update **Description** The date until when the rebate partner special pricing term benefit is effective.
`EffectiveStartDate`	**Type** dateTime **Properties** Create, Filter, Nillable, Sort, Update **Description** The date from when the rebate partner special pricing term benefit is effective.
`IsDiscountByPercent`	**Type** boolean **Properties** Create, Defaulted on create, Filter, Group, Sort, Update **Description** Indicates whether the discount is given by percentage (`true`) or amount (`false`). The default value is `false`.
`LastReferencedDate`	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** The most recent date on which a user referenced this record.
`LastViewedDate`	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** The most recent date on which a user viewed this record.
`MaximumQuantity`	**Type** double **Properties** Create, Filter, Nillable, Sort, Update **Description** The maximum quantity of product sold that is eligible for the rebate partner special pricing term benefit.

Field	Details
`MinimumQuantity`	**Type** double **Properties** Create, Filter, Nillable, Sort, Update **Description** The minimum quantity of product sold that is eligible for the rebate partner special pricing term benefit.
`Name`	**Type** string **Properties** Create, Filter, Group, idLookup, Sort, Update **Description** The name of the rebate partner special pricing term benefit.
`RebatePartnerSpecialPrcTrmId`	**Type** reference **Properties** Create, Filter, Group, Sort **Description** The rebate partner special pricing term that's associated with the rebate partner special pricing term benefit. This field is a relationship field. **Relationship Name** RebatePartnerSpecialPrcTrm **Relationship Type** Lookup **Refers To** RebatePartnerSpecialPrcTrm
`ReferencePricePerUnit`	**Type** currency **Properties** Create, Filter, Nillable, Sort, Update **Description** The reference price per unit that’s applicable for the rebate partner special pricing term benefit.

ReceivedDocument

Allows partners to upload .CSV document. This object is available in API version 51.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
`FailedRecordsFileId`	**Type** reference **Properties** Filter, Group, Nillable, Sort, Update **Description** Lookup to the error file.
`Name`	**Type** string **Properties** Create, Filter, Group, idLookup, Sort, Update **Description** The name of the received document.
`Status`	**Type** picklist **Properties** Create, Filter, Group, Restricted picklist, Sort, Update **Description** Current stage of processing. Possible values are: • `APPROVED`—Approved • `ARCHIVED`—Archived • `DRAFT`—Draft • `IN_REVIEW`—In Review • `OBSOLETE`—Obsolete • `SUBMITTED`—Submitted • `SUPERSEDED`—Superseded

Field	Details
`UnprocessedRecordsFileId`	**Type** reference **Properties** Filter, Group, Nillable, Sort, Update **Description** Lookup to the skipped file.

TransactionJournal

The transactions that need to be processed for a rebate program. For example, order line for 1000 units of $1200 for member ABC enterprises. This object is available in API version 51.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
`AccountId`	**Type** reference **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** The account for which the transaction is made. This is a relationship field. **Relationship Name** Account **Relationship Type** Lookup **Refers To** Account
`ActivityDate`	**Type** dateTime **Properties** Create, Filter, Sort, Update

Field	Details
	Description Date the transaction happened.
`ErrorDescription`	**Type** string **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** Description of the error in the journal record.
`JournalDate`	**Type** dateTime **Properties** Create, Filter, Nillable, Sort, Update **Description** Date the transaction was received.
`MemberId`	**Type** reference **Properties** Create, Filter, Group, Sort, Update **Description** Polymorphic key to RebateProgramMember or LoyaltyMember.
`ProductId`	**Type** reference **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** The product associated with the transaction.
`Quantity`	**Type** double **Properties** Create, Filter, Nillable, Sort, Update **Description** The order quantity.
`QuantityUnitOfMeasureId`	**Type** reference

Field	Details
	Properties Create, Filter, Group, Nillable, Sort, Update **Description** The unit of measure associated with the record. This is a relationship field. **Relationship Name** QuantityUnitOfMeasure **Relationship Type** Lookup **Refers To** UnitOfMeasure
`RebatePgmReferenceNumber`	**Type** string **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** The reference number of the rebate program. This field is available in API version 52.0 and later.
`ShipToAccountId`	**Type** reference **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** The account for which the additional benefit is being offered for ship and debit program. This field is available in API version 52.0 and later. This is a relationship field. **Relationship Name** ShipToAccount **Relationship Type** Lookup **Refers To** Account
`TransactionAmount`	**Type** currency **Properties** Create, Filter, Nillable, Sort, Update **Description** The total amount of the transaction.

Field	Details
`UsageType`	**Type** picklist **Properties** Create, Filter, Group, Nillable, Restricted picklist, Sort **Description** Type of journal record. Possible values are: • `Loyalty` • `Rebates`

UnitOfMeasureConversion

Represents the information used to convert a measurement value from a unit of measure to another. This object is available in API version 52.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
`ConversionFactor`	**Type** double **Properties** Create, Filter, Sort, Update **Description** The conversion factor for the conversion.
`FromUnitOfMeasureId`	**Type** reference **Properties** Create, Filter, Group, Sort, Update **Description** The unit of measure to be converted. This is a relationship field. **Relationship Name** FromUnitOfMeasure

Field	Details
	Relationship Type Lookup **Refers To** UnitOfMeasure
`ToUnitOfMeasureId`	**Type** reference **Properties** Create, Filter, Group, Sort, Update **Description** The unit of measure to be converted to. This is a relationship field. **Relationship Name** ToUnitOfMeasure **Relationship Type** Lookup **Refers To** UnitOfMeasure

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

UnitOfMeasureConversionFeed
Feed tracking is available for the object.

UnitOfMeasureConversionHistory
History is available for tracked fields of the object.

UnitOfMeasureConversionOwnerSharingRule
Sharing rules are available for the object.

UnitOfMeasureConversionShare
Sharing is available for the object.

REBATE MANAGEMENT STANDARD INVOCABLE ACTIONS

Add members to a rebate program, calculate rebate amounts and upsert payouts, generate rebate payout periods, process rebate calculation batch jobs, and process a rebate CSV file for Rebate Management. For more information on standard invocable actions, see **REST API Developer Guide** and **Actions Developer Guide**.

Add Rebate Member List

Add a list of members to a rebate program. The list of rebate members is based on the accounts in the specified account list view.

Calculate Rebate Amount and Upsert Payout

Calculate the rebate amount and upsert the rebate payout for the specified aggregate record.

Calculate Projected Rebate Amount

Calculate the projected rebate amount for rebate types associated with a specified transaction reference ID.

Generate Rebate Payout Periods

Generate payout periods for a rebate program based on the frequency specified in the program.

Get Benefit and Calculate Rebate Amount

Get the details benefits and optionally calculate the rebate amount for the specified aggregate record.

Get Eligible Program Rebate Types

Retrieve the eligible program rebate types for a mapped object.

Process Program Rebate Type Products

Insert or delete the records in the Program Rebate Type Product object. The inserted products participate as inclusion or exclusion as defined in the Product Filter Type option on Program Rebate Type.

Process Rebate Batch Calculation Jobs

Process a rebate batch calculation job from the Data Processing Engine.

Process Rebate CSV Files

Process an uploaded CSV file using Bulk API 2.0 and convert the file's data into records in the target object.

Upsert Custom Rebate Payout

Upsert the custom calculated rebate payout for a specified aggregate record.

Add Rebate Member List

Add a list of members to a rebate program. The list of rebate members is based on the accounts in the specified account list view.

This object is available in API version 51.0 and later for users with Rebate Management license.

Supported REST HTTP Methods

URI

`/services/data/v`***XX.X***`/actions/standard/addRebateMemberList`

Formats

JSON, XML

HTTP Methods

GET, HEAD, POST

Authentication

`Authorization: Bearer token`

Inputs

Input	Details
`listViewId`	**Type** string **Description** Required. The ID of the account list that's specified to add rebate members.
`notificationPreference`	**Type** string **Description** A text value that indicates whether notifications should be sent for successful and unsuccessful rebate member additions. Valid values include BlockSuccessEmails, BlockFailureEmails, BlockAllEmails, and AllowAllNotifications.
`rebateProgramId`	**Type** string **Description** Required. The ID of the rebate program to which specified account list view is added.
`status`	**Type** string **Description** Required. The status of the rebate members for the accounts in the specified list view.

Usage

Sample Request

```
{
   "inputs":[{
   "listViewId":"a07B0000007qbQOIAY",
   "rebateProgramId":"x18Z1236547qbQOBCV",
   "status": "Active"
   "notificationPreference": "BlockAllEmails"
}]
}
```

Sample Response

```
{
   "isSuccess":true
}
```

Calculate Rebate Amount and Upsert Payout

Calculate the rebate amount and upsert the rebate payout for the specified aggregate record.

This object is available in API version 51.0 and later for users with Rebate Management license.

Supported REST HTTP Methods

URI

`/services/data/v`***`XX.X`***`/actions/standard/calculateRebateAmountAndUpsertPayout`

Formats

JSON, XML

HTTP Methods

GET, HEAD, POST

Authentication

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

Inputs

Input	Details
`aggregateId`	**Type** ID **Description** Required. The ID of the aggregate record for which you want to calculate the rebate amount and upsert the rebate payout.
`payoutMemberFieldName`	**Type** String **Description** The name of the field that contains the rebate program member ID for which to perform the payout calculations.

Outputs

Output	Details
programRebateTypePayoutSourceId	**Type** ID **Description** The ID of the program rebate type payout source record, which is generated if the request is successful.

Usage

Sample Request

```
{
   "inputs":[
      {"aggregateId":"a00xx000000cCtxAAE"},
      {"aggregateId":"a00xx000000cCvZAAU"},
      {"aggregateId":"a00xx000000cCxBAAU"}
   ]
}
```

Sample Response

```
{
   "errors":null,
   "isSuccess":true,
   "outputValues":{
      "programRebateTypePayoutSourceId":"0ntxx00004000002AAA"
   }
}
```

Calculate Projected Rebate Amount

Calculate the projected rebate amount for rebate types associated with a specified transaction reference ID.

This object is available in API version 54.0 and later.

Supported REST HTTP Methods

URI

`/services/data/v`***XX.X***`/actions/standard/calculateProjectedRebateAmount`

Formats

JSON

HTTP Methods

POST

Authentication

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

Inputs

Input	Details
`programRebateTypeIds`	**Type** string **Description** Required. The list of program rebate type IDs to calculate the projected rebate amounts for.
`referenceRecordId`	**Type** string **Description** Required. The transaction reference record ID to calculate the projected rebate amounts for.

Outputs

Output	Details
`calculateProjectedRebateAmount`	**Type** string **Description** The calculated rebate amount details for the program rebate types associated with the transaction reference record. This is an Apex output type that contains nested arrays in JSON format. `calculateProjectedRebateAmount` contains an array of the `CalculateProjectedRebateAmountOutputRepresentation` Apex class. This output representation contains an array of the `ProjectedRebateAmountCalc` fields, while an array of `DetailedProjectedRebateAmountCalc` is contained in `ProjectedRebateAmountCalc`.

Note: The fields `aggregateId`, `benefitQualifierValue`, `benefitTier`, `measureFieldValue`, `qualifiedTiers`, `rebateTypeBenefit`, `totalProjectedRebateAmount`, `payoutPeriod`, `projectedRebateAmount` can appear in the output list. These fields must be ignored because they aren't supported in the flow for this invocable action. These fields will be deprecated in API version 55.0.

Usage

JSON Sample Request

```
{
   "inputs":[ {
      "programRebateTypeIds": ["0hvRM0000004CUt"],
      "referenceRecordId": "801RM00000083TeYAI"
```

```
    } ]
}
```

JSON Sample Response

```
{
  "projectedRebateAmountCalc": [
    {
      "programRebateTypeId": "0hvRM0000004CUt",
      "referenceRecordId": "801RM00000083TeYAI",
      "totalProjectedRebateAmount": 1088000,
      "detailedProjectedRebateAmountCalc": [
        {
          "benefitQualifierValue": 500,
          "benefitTier": "0huRM0000004CRC",
          "rebateTypeBenefit": "5",
          "measureFieldValue": 160000,
          "payoutPeriod": "0i7RM0000004CV7",
          "projectedRebateAmount": 800000,
          "qualifiedTiers": [
            "0huRM0000004CRC"
          ]
        },
        {
          "benefitQualifierValue": 240,
          "benefitTier": "0huRM0000004CRC",
          "rebateTypeBenefit": "5",
          "measureFieldValue": 57600,
          "payoutPeriod": "0i7RM0000004CV7",
          "projectedRebateAmount": 288000,
          "qualifiedTiers": [
            "0huRM0000004CRC"
          ]
        }
      ]
    }
  ]
}
```

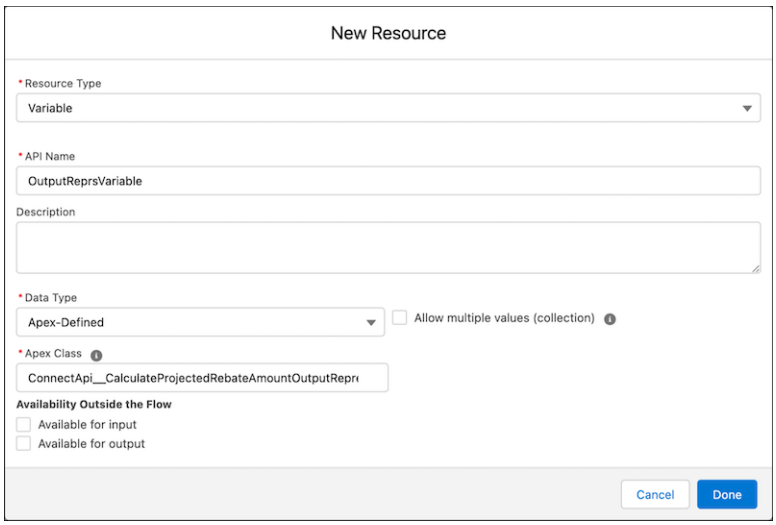
Details of the Apex output type

- `projectedRebateAmountCalc`—An array of projected rebate amounts. Each item specifies the projected rebate amount for a passed rebate type and the reference object.
  - `programRebateTypeId`—Specifies the program rebate type ID whose projected rebate amounts are calculated in this item.
  - `referenceRecordId`—Specifies the transaction reference record ID based on which the projected rebate amounts are calculated.
  - `totalProjectedRebateAmount`—Specifies the total projected rebate amount for this program rebate type and reference record.
  - `detailedProjectedRebateAmountCalc`—An array with each item corresponding to each aggregate record that is found for a combination of rebate type and reference object. For example, if the rebate is computed differently for every product using a benefit mapping and the reference object (order) has two order lines, this collection has two aggregates, one for each product.

- `benefitQualifierValue`—Specifies the total benefit qualifier value calculated from both the aggregate record and the reference object.
- `benefitTier`—Specifies the benefit tier ID that is determined based on the benefit qualifier value.
- `rebateTypeBenefit`—Specifies the applied benefit that is granted based on the benefit tier identified.
- `measureFieldValue`—Specifies the total measure value calculated from both the aggregate record and the reference object.
- `payoutPeriod`—Specifies the rebate program payout period that is determined based on the mapped member ID and activity date. The mapping between reference object and transaction journal is used here. See Map Fields Between Target Transaction Object and Transaction Journal.
- `projectedRebateAmount`—Specifies the calculated projected rebate amount based on the benefit tier and the measure field value.
- `qualifiedTiers`—List of the eligible benefit tiers that have already been met.

Usage

You must configure the output parameter for the Calculate Projected Rebate Amount Flow Action in Flow Builder. The following screen illustrates how to use the output parameter in the Flow Builder.



Generate Rebate Payout Periods

Generate payout periods for a rebate program based on the frequency specified in the program.

This object is available in API version 51.0 and later for users with Rebate Management license.

Supported REST HTTP Methods

URI

`/services/data/v***XX.X***/actions/standard/generateRebatePayoutPeriods`

Formats
JSON, XML

HTTP Methods
GET, HEAD, POST

Authentication
`Authorization: Bearer token`

Inputs

Input	Details
`rebateProgramId`	**Type** ID **Description** Required. The ID of the rebate program record for which you want to generate payout periods.

Outputs

Output	Details
`rebateProgramPayoutPeriods`	**Type** string **Description** The list of rebate program payout period IDs upserted if the request is successful.

Usage

Sample Request

```
{
   "inputs":[{"rebateProgramId":"0lcxx00004000002AAA"},
      {"rebateProgramId":"0lcxx00004000002AAA"}
   ]
}
```

Sample Response

```
{
   "errors":null,
   "isSuccess":true,
   "RebateProgramPayoutPeriodIds":["0ntxx00004000002AAA","0ntxx00004000002AAB"]
},{
   "errors":null,
   "isSuccess":true,
   "RebateProgramPayoutPeriodIds":["0ntxx00004000002AAA","0ntxx00004000002AAB"]
}
```

Get Benefit and Calculate Rebate Amount

Get the details benefits and optionally calculate the rebate amount for the specified aggregate record.

This object is available in API version 51.0 and later for users with Rebate Management license.

Supported REST HTTP Methods

URI

`/services/data/v`***`XX.X`***`/actions/standard/getBenefitAndCalculateRebateAmount`

Formats

JSON, XML

HTTP Methods

GET, HEAD, POST

Authentication

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

Inputs

Input	Details
`aggregateId`	**Type** ID **Description** Required. The ID of the aggregate record for which you want to calculate the rebate amount.
`payoutMemberFieldName`	**Type** ID **Description** The name of the field that contains the rebate program member ID for which to perform the payout calculations.
`skipRebateCalc`	**Type** Boolean **Description** Indicates whether to skip calculation of the rebate amount.

Outputs

Output	Details
`accuralRate`	**Type** Double

Output	Details
	Description The accrual rate if the request is successful.
`accruedAmount`	**Type** Double **Description** The accrued amount calculated if the request is successful.
`aggregateId`	**Type** ID **Description** The ID of the aggregate record for which you want to calculate the rebate amount.
`benefitId`	**Type** ID **Description** The ID of the applied Program Rebate Type Benefit record if the request is successful.
`payoutMemberFieldName`	**Type** String **Description** The name of the field that contains the rebate program member ID for which to perform the payout calculations.
`qualifyingBenefits`	**Type** string **Description** The comma-separated list of IDs of the qualified Program Rebate Type Benefit records if the request is successful.
`rebateAmount`	**Type** double **Description** The rebate amount calculated if the request is successful.

Usage

Sample Request

```
{
   "inputs":[
      {"AggregateId": "a00xx000000cCtxAAE",
      "SkipRebateCalc": false},
```

```
      {"AggregateId": "a00xx000000cCvZAAU",
      "SkipRebateCalc": false},
      {"AggregateId": "a00xx000000cCxBAAU",
      "SkipRebateCalc": false}
   ]
}
```

Sample Response

```
{
   "errors":null,
   "isSuccess":true,
   "outputValues":{
      "AggregateId":"0ntxx00004000002AAA",
      "BenefitId":"0ntxx00004000002AAA",
      "RebateAmount":"2000"
   }
}
```

Get Eligible Program Rebate Types

Retrieve the eligible program rebate types for a mapped object.

For more information about how to use the `getEligibleProgramRebateTypes` action in Flow Builder, see Configure the Output Parameter for the Get Eligible Program Rebate Types Flow Action in Flow Builder in Salesforce Help.

This object is available in API version 52.0 and later.

Supported REST HTTP Methods

URI

`/services/data/v`***`XX.X`***`/actions/standard/getEligibleProgramRebateTypes`

Formats

JSON

HTTP Methods

POST

Authentication

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

Inputs

Input	Details
`appliedRebateTypesLimit`	**Type** integer **Description** The maximum number of search results to be returned for the applied rebate types. The input value is passed through to the output parameter.

Input	Details
	This field is available in API version 54.0 and later.
`appliedRebateTypesOffsetInput`	**Type** integer **Description** The input offset for the applied rebate types search results plus the number of applied rebates types processed. Used to process applied rebate types in a loop. When all applied rebate types have been processed, zero is returned. This field is available in API version 54.0 and later.
`limit`	**Type** integer **Description** The maximum number of search results to be returned for the eligible rebate types.
`measureTypeFilter`	**Type** array **Description** Filters the eligible rebate types by the specified measure type.
`offset`	**Type** integer **Description** The offset for the eligible rebate types search results. The offset parameter specifies the end of the last batch retrieved. For example, to retrieve 100 rebate types at a time: • Request the first 100 with an offset of 0. • Request the second 100 with an offset of 100. • Request the third 100 with an offset of 200.
`queryOnlyIntegrableProgramRebateTypes`	**Type** boolean **Description** Indicates whether only integrable program rebate types must be queried.
`referenceObjectIds`	**Type** array **Description** Required. The list of IDs of the mapped object for which to get the eligible rebate types.
`searchByRebateTypeName`	**Type** string

Input	Details
	Description Searches the eligible rebate types by the specified name.

Outputs

Output	Details
`resultantRebateTypes`	**Type** string **Description** The list of eligible and applied rebate type IDs for the mapped object. Use with API calls.
`resultantRebateTypesForFlow`	**Type** string **Description** The list of eligible and applied rebate type IDs for the mapped object. Use with Flow Builder. This field is available in API version 54.0 and later.

Usage

JSON Sample Request

```
{
   "inputs":[{
      "referenceObjectIds":["0lcxx00004000002AAA","0lcxx00004000002AAB"],
      "queryOnlyIntegratableProgramRebateTypes": true,
      "limit": 10,
      "offset": 5
   }]
}
```

JSON Sample Response

```
{
   resultantRebateTypes: {
      "isSuccess":true,
      "programRebateTypeIDs":[
          {ReferenceObjectID: "0lcxx00004000002AAA", eligibleRebateTypes:
[0lcxx00004000003AAA,0lcxx00004000003AAB], appliedRebateTypes:
[0lcxx00004000004AAA,0lcxx00004000004AAB]},
          {ReferenceObjectID: "0lcxx00004000002AAB", eligibleRebateTypes:
[0lcxx00004000003AAC,0lcxx00004000003AAD], appliedRebateTypes:
[0lcxx00004000005AAA,0lcxx00004000005AAB]}
      ],
      "limit": 10,
      "offset": 13,
```

```
      "appliedRebateTypesLimit": 5,
      "appliedRebateTypesOffset" 9,
      "errors":null
   },
   resultantRebateTypesV2:{
      "isSuccess":true,
      "programRebateTypeIDs":[
          {ReferenceObjectID: "01cxx00004000002AAA", eligibleRebateTypes:
[01cxx00004000003AAA,01cxx00004000003AAB], appliedRebateTypes:
[01cxx00004000004AAA,01cxx00004000004AAB]},
          {ReferenceObjectID: "01cxx00004000002AAB", eligibleRebateTypes:
[01cxx00004000003AAC,01cxx00004000003AAD], appliedRebateTypes:
[01cxx00004000005AAA,01cxx00004000005AAB]}
      ],
      "limit": 10,
      "offset": 13,
      "appliedRebateTypesLimit": 5,
      "appliedRebateTypesOffset": 9,
      "errors":null
   }
}
```

Process Program Rebate Type Products

Insert or delete the records in the Program Rebate Type Product object. The inserted products participate as inclusion or exclusion as defined in the Product Filter Type option on Program Rebate Type.

For more information about how the processProgramRebateTypeProducts action inserts or deletes records, see Create Payout Calculation Flows with Flow Actions in Salesforce Help.

This object is available in API version 53.0 and later.

Supported REST HTTP Methods

URI

`/services/data/v`***`XX.X`***`/actions/standard/processProgramRebateTypeProducts`

Formats

JSON

HTTP Methods

POST

Authentication

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

Inputs

Input	Details
`operationType`	**Type** string

Input	Details
	Description Required. The type of operation to be performed on the program rebate type product records. Possible values are: • `Delete`—Deletes all program rebate type products for the specified program rebate type ID. • `Insert`—Inserts all products as program rebate type products, preserves the existing ones.
`productListViewId`	**Type** string **Description** The ID of the product list view containing the products to be added to the inclusion or exclusion list of this program rebate type. Note: This field isn't used when the type of operation is `Delete`.
`programRebateTypeId`	**Type** string **Description** Required. The ID of the program rebate type for which the records are being processed.

Usage

JSON Sample Request

```
{
   "inputs":[{
      "operationType" : "insert",
      "productListViewId": "00BHr00000Q6zqNMAR",
      "programRebateTypeId": "0hvHr000000HDQQIA4"
   }]
}
```

Process Rebate Batch Calculation Jobs

Process a rebate batch calculation job from the Data Processing Engine.

This object is available in API version 51.0 and later for users with Rebate Management license.

Supported REST HTTP Methods

URI

`/services/data/v`***XX.X***`/actions/standard/processRebatesBatchCalculationJob`

Formats

JSON, XML

HTTP Methods
GET, HEAD, POST

Authentication
`Authorization: Bearer token`

Inputs

Input	Details
`calcDefinitionId`	**Type** ID **Description** Required. The ID of the data processing engine definition record that you want to process.
`rebateProgramIds`	**Type** array **Description** The list of IDs of the rebate program records that the custom batch calculation job processes.
`programRebateTypeIds`	**Type** array **Description** The list of IDs of the program rebate type records that the custom batch calculation job processes.
`rebateProgramPayoutPeriodIds`	**Type** array **Description** The list of IDs of the rebate program payout period records that the custom batch calculation job processes.
`rebateProgramMemberIds`	**Type** array **Description** The list of IDs of the rebate program member records that the custom batch calculation job processes.

Outputs

Output	Details
`batchJobId`	**Type** ID

Output	Details
	Description The ID of the generated batch job record after the request is successful.
`accepted`	**Type** boolean **Description** Indicates whether the rebate batch calculation job is accepted.

Usage

Sample Request

```
{
   "inputs":[{
      "calcDefinitionId": "0lcxx00004000002AAA",
      "rebateProgramIds": ["0htxx00004000002AAA","0htxx00004000002bAA"],
      "programRebateTypeIds": ["0htxx00004000002AAA","0htxx00004000002bAA"],
      "rebateProgramPayoutPeriodIds": ["0htxx00004000002AAA","0htxx00004000002bAA"],
      "rebateProgramMemberIds": ["0htxx00004000002AAA","0htxx00004000002bAA"]
   }]
}
```

Sample Response

```
{
   "errors":null,
   "isSuccess":true,
   "outputValues":{
      "batchJobId":"0ntxx00004000002AAA",
      }
}
```

Process Rebate CSV Files

Process an uploaded CSV file using Bulk API 2.0 and convert the file's data into records in the target object.

This object is available in API version 51.0 and later for users with Rebate Management license.

Supported REST HTTP Methods

URI

`/services/data/v`***XX.X***`/actions/standard/rebatesProcessCSV`

Formats

JSON, XML

HTTP Methods

GET, HEAD, POST

Authentication

```
Authorization: Bearer token
```

Inputs

Input	Details
`recordId`	**Type** ID **Description** Required. The ID of the received document record that includes the CSV file to be converted.
`entityAPIName`	**Type** string **Description** Required. The API name of the target object that receives the processed CSV file records.
`lineEnding`	**Type** string **Description** Optional. The line ending request header that specifies whether the line endings can be read either as line feeds (LFs) or as line feeds (CRLFs) for fields of type Text Area and Text Area (Long). The default value is `LF`.

Usage

Sample Request

```
{
   "inputs":[{
      "recordId": "0io4S000000TPsyQAG",
      "entityAPIName" : "TransactionJournal"
   }]
}
```

Sample Response

```
{
   "errors":null,
   "isSuccess":true,
}
```

Upsert Custom Rebate Payout

Upsert the custom calculated rebate payout for a specified aggregate record.

This object is available in API version 51.0 and later for users with Rebate Management license.

Supported REST HTTP Methods

URI

`/services/data/v***XX.X***/actions/standard/upsertCustomRebatePayout`

Formats

JSON, XML

HTTP Methods

GET, HEAD, POST

Authentication

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

Inputs

Input	Details
`accuralRate`	**Type** Double **Description** The accrual rate used to calculate the accrued amount for the generated program rebate type payout source record.
`accruedAmount`	**Type** Double **Description** The accrued amount calculated for the generated program rebate type payout source record.
`aggregateId`	**Type** ID **Description** Required. The ID of the aggregate record for which you want to upsert the rebate payout.
`benefitId`	**Type** ID **Description** The ID of the qualified program rebate type benefit record for which you want to upsert the rebate payout.
`payoutMemberFieldName`	**Type** String **Description** The name of the field that contains the rebate program member ID for which to perform the payout calculations.

Input	Details
rebateAmount	**Type** double **Description** Required. The rebate amount for which you want to upsert the rebate payout.

Outputs

Output	Details
programRebateTypePayoutSourceId	**Type** ID **Description** The ID of the program rebate type payout source record, which is generated after the request is successful.

Usage

Sample Request

```
{
   "inputs":[{
      "AggregateId":"0lcxx00004000002AAA",
      "BenefitId":"0htxx00004000002AAA",
      "RebateAmount":"2000"
   }]
}
```

Sample Response

```
{
   "errors":null,
   "isSuccess":true,
   "outputValues":{
      "programRebateTypePayoutSourceId":"0ntxx00004000002AAA",
   }
}
```

REBATE MANAGEMENT METADATA API TYPES

Metadata API enables you to access some types and feature settings that you can customize in the user interface. For more information about Metadata API and to find a complete reference of existing metadata types, see **Metadata API Developer Guide**.

ObjectHierarchyRelationship
Represents an organization's custom field mappings between a reference object and eligible rebate types. Fields can be mapped from Opportunity, OpportunityLineItem, Quote, QuoteLineItem, and Contract to TransactionJournal.

ObjectHierarchyRelationship

Represents an organization's custom field mappings between a reference object and eligible rebate types. Fields can be mapped from Opportunity, OpportunityLineItem, Quote, QuoteLineItem, and Contract to TransactionJournal.

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

ObjectHierarchyRelationship components have the suffix `ObjectHierarchyRelationship.settings` and are stored in the `ObjectHierarchyRelationship` folder.

Version

ObjectHierarchyRelationship components are available in API version 51.0 and later.

Fields

Field Name	Description
`childObjectMapping`	**Field Type** ObjectMapping **Description** Set of `inputObject`, `mappingFields`, and `outputObject` entries.
`inputObjRecordsGrpFieldName`	**Field Type** string **Description** The field name in the input object used to group the records. This field is available in API version 55.0 and later.
`mappingType`	**Field Type** ObjHierarchyMappingType (enumeration of type string)

Field Name	Description
	Description Specifies the type of relationship between two objects. Valid values are: • `ChildToChild` • `ParentToChild` • `ParentToParent` • `Support` This field is available in API version 55.0 and later.
`masterLabel`	**Field Type** string **Description** Master label name of the mapping definition.
`outputPntRelationshipFieldName`	**Field Type** string **Description** The field name that defines the relationship between a parent and child for the output object. This field is available in API version 55.0 and later.
`parentObjectMapping`	**Field Type** ObjectMapping **Description** Required. Set of `inputObject`, `mappingFields`, and `outputObject` entries.
`parentRecord`	**Field Type** string **Description** The parent record for this object hierarchy relationship. This field is available in API version 55.0 and later.
`parentRelationshipFieldName`	**Field Type** string **Description** Name of the field that defines the relationship between the parent and child.
`usageType`	**Field Type** MappingUsageType (enumeration of type string) **Description** Required.

Field Name	Description
	Name of the usage type of an object hierarchy relationship. Valid value is: • `ConvertToSalesAgreement` • `CLMFieldMapping` • `EligibleProgramRebateType` • `MapJournalToMemberAggregate` • `TransformationMapping`

ObjectMapping

Represents a set of `inputObject`, `mappingFields`, and `outputObject` entries.

Fields

Field Name	Description
`inputObject`	**Field Type** string **Description** Required. Name of the input object type containing the source fields for mapping. For example, Opportunity or OpportunityLineItem. Note: You can also use a custom object as the input object type.
`mappingFields`	**Field Type** ObjectMappingField **Description** Mapping of source object input fields to target object.
`outputObject`	**Field Type** string **Description** Required. Name of the output object type receiving data conversion. For example, TransactionJournal.

ObjectMappingField

Represents a set of `inputField` and `outputField` entries.

Fields

Field Name	Field Type	Description
`inputField`	string	**Field Type** string **Description** Required. Field in the object specified by the `inputObject` field in ObjectMapping on page 93. This field is mapped to the field in `outputField`, which is a field in the object specified by the `outputObject` field in ObjectMapping on page 93. Note: You can also use custom fields as the input field type.
`outputField`	string	**Field Type** string **Description** Required. Field in the object specified by the `outputObject` field in ObjectMapping on page 93. This field is mapped to the field name in `inputField`, which is a field in the object specified by the `inputObject` field in ObjectMapping on page 93. Note: You can also use custom fields as the output field type.

Declarative Metadata Sample Definition

The following are examples of a ObjectHierarchyRelationship component.

Example of mapping between Opportunity and Transaction Journal.

```
<?xml version="1.0" encoding="UTF-8"?>
<ObjectHierarchyRelationship xmlns="http://soap.sforce.com/2006/04/metadata">
   <parentObjectMapping>
      <inputObject>Opportunity</inputObject>
      <outputObject>TransactionJournal</outputObject>
      <mappingFields>
        <inputField>AccountId</inputField>
        <outputField>MemberId</outputField>
      </mappingFields>
      </mappingFields>
   </parentObjectMapping>
   <childObjectMapping>
   <inputObject>OpportunityLineItem</inputObject>
      <outputObject>TransactionJournal</outputObject>
      <mappingFields>
```

```
        <inputField>Product2Id</inputField>
        <outputField>ProductId</outputField>
      </mappingFields>
      <mappingFields>
        <inputField>Quantity</inputField>
        <outputField>Quantity</outputField>
      </mappingFields>
   </childObjectMapping>
   <usageType>EligibleProgramRebateType</usageType>
</ObjectHierarchyRelationship>
```

Example of mapping between Quote and Transaction Journal.

```
<?xml version="1.0" encoding="UTF-8"?>
<ObjectHierarchyRelationship xmlns="http://soap.sforce.com/2006/04/metadata">
   <parentObjectMapping>
      <inputObject>Quote</inputObject>
      <outputObject>TransactionJournal</outputObject>
      <mappingFields>
        <inputField>AccountId</inputField>
        <outputField>MemberId</outputField>
      </mappingFields>
   </parentObjectMapping>
   <childObjectMapping>
      <inputObject>QuoteLineItem</inputObject>
      <outputObject>TransactionJournal</outputObject>
     <mappingFields>
        <inputField>Product2Id</inputField>
        <outputField>ProductId</outputField>
      </mappingFields>
      <mappingFields>
        <inputField>Quantity</inputField>
        <outputField>Quantity</outputField>
      </mappingFields>
   </childObjectMapping>
   <usageType>EligibleProgramRebateType</usageType>
</ObjectHierarchyRelationship>
```

Example of mapping between Contract and Transaction Journal.

```
<?xml version="1.0" encoding="UTF-8"?>
<ObjectHierarchyRelationship xmlns="http://soap.sforce.com/2006/04/metadata">
   <parentObjectMapping>
      <inputObject>Contract</inputObject>
      <outputObject>TransactionJournal</outputObject>
      <mappingFields>
        <inputField>AccountId</inputField>
        <outputField>MemberId</outputField>
      </mappingFields>
    </parentObjectMapping>
    <usageType>EligibleProgramRebateType</usageType>
</ObjectHierarchyRelationship>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
      <members>*</members>
      <name>ObjectHierarchyRelationship</name>
    </types>
    <version>51.0</version>
</Package>
```

Usage

Use the `deploy()` call to deploy metadata with a .zip file. Every .zip file contains a project manifest, a file that's named package.xml, and a set of directories that contain the components. The manifest file defines the components that you're trying to retrieve or deploy in the .zip file. The manifest also defines the API version that's used for the deployment or retrieval. For more information on the .zip file, deploying, and retrieving the metadata, see Deploying and Retrieving Metadata with the Zip File. You can also deploy and retrieve the metadata API using Postman.

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

DATA PROCESSING ENGINE, BATCH MANAGEMENT, AND MONITOR WORKFLOW SERVICES

Data Processing Engine and Batch Management help automate your business processes. Use objects, APIs, Platform Events, and invocable actions to define, run, and review Data Processing Engine definitions and Batch Management jobs.

Here's how both these features can automate your business processes:

- Data Processing Engine: Transform data that's available in your Salesforce org and write back the transformation results as new or updated records. You can transform the data for standard and custom objects.
- Batch Management: Automate the processing of records in scheduled flows. You can process a high volume of standard and custom object records.

Once a Data Processing Engine definition is run or a Batch Management job is run, you can view the progress of the run and the results of the run using Monitor Workflow Services.

EDITIONS

Available in: Lightning Experience

Available in: Data Processing Engine is available with Enterprise, Unlimited, and Performance Editions with the Financial Services Cloud, Loyalty Management, Manufacturing Cloud, Rebate Management, Accounting Subledger, or Provider Search in Health Cloud.

Available in: Batch Management is available with Enterprise, Unlimited, and Performance Editions with the Loyalty Management, Manufacturing Cloud, Rebate Management, or Accounting Subledger

Data Model

Data Processing Engine, Batch Management, and Monitor Workflow Servics share a data model. Let's learn about the objects and relationships in this shared data model.

Common Tooling API Object

BatchJobDefinition is a common Tooling API object that is shared between Data Processing Engine and Batch Management.

Common Platform Event

Batch Management jobs and Data Processing Engine definitions are run using invocable actions in Flows. Use the BatchJobStatusChanged event to notify subscribers after a Batch Management job or a Data Processing Engine definition is processed in a flow.

Common Business APIs

Common Business APIs are RESTful APIs that are sometimes available as Apex classes and methods.

Data Processing Engine

Transform data that's available in your Salesforce org and write back the transformation results as new or updated records. You can transform the data for standard and custom objects. Data Processing Engine consists of a Tooling API object, a standard object, a Metadata API, and an invocable action. You can use these to view, create, edit, and run Data Processing Engine definitions.

Batch Management

Automate the processing of records in scheduled flows. You can process a high volume of standard and custom object records. Batch Management consists of three Tooling API objects, a standard object, a Metadata API, and an invocable action. You can use these resources to view, create, edit, and run Batch Management jobs.

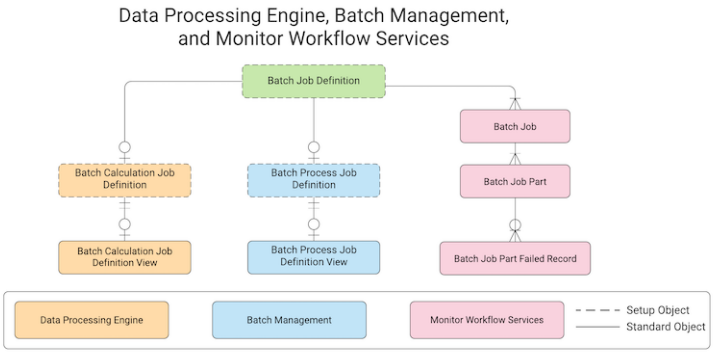
Monitor Workflow Services

The Montior Workflow Services standard objects can be used to track the run of Data Processing Engine definitons and Batch Management jobs. During a run, you can view details about each part that the run is broken down into. After the run is complete, you can view its status and the records which weren't processed during the run.

Data Model

Data Processing Engine, Batch Management, and Monitor Workflow Servics share a data model. Let's learn about the objects and relationships in this shared data model.

Here's the data model:



EDITIONS

Available in: Lightning Experience

Available in: Data Processing Engine is available with Enterprise, Unlimited, and Performance Editions with the Financial Services Cloud, Loyalty Management, Manufacturing Cloud, Rebate Management, Accounting Subledger, or Provider Search in Health Cloud.

Available in: Batch Management is available with Enterprise, Unlimited, and Performance Editions with the Loyalty Management, Manufacturing Cloud, Rebate Management, or Accounting Subledger

Common Tooling API Object

BatchJobDefinition is a common Tooling API object that is shared between Data Processing Engine and Batch Management.

Tooling API objects let you interact with metadata for declarative development. For example, you can create your own version of Setup.

BatchJobDefinition
Represents the definition of a batch job. This object is available in API version 51.0 and later.

EDITIONS

Available in: Lightning Experience

Available in: Data Processing Engine is available with Enterprise, Unlimited, and Performance Editions with the Financial Services Cloud, Loyalty Management, Manufacturing Cloud, Rebate Management, Accounting Subledger, or Provider Search in Health Cloud.

Available in: Batch Management is available with Enterprise, Unlimited, and Performance Editions with the Loyalty Management, Manufacturing Cloud, Rebate Management, or Accounting Subledger

BatchJobDefinition

Represents the definition of a batch job. This object is available in API version 51.0 and later.

Note: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Supported SOAP API Calls

`describeSObjects()`, `query()`, `retrieve()`

Supported REST API Methods

`GET, HEAD, Query`

Fields

Field	Details
`Description`	**Type** textarea **Properties** Filter, Group, Nillable, Sort **Description** The description of the batch job definition.
`DeveloperName`	**Type** string **Properties** Filter, Group, Sort **Description** The developer name of the batch job.
`Language`	**Type** picklist

Field	Details
	Properties Filter, Group, Restricted picklist, Sort **Description** The language in which the batch job is created. Possible values are: • `af`—Afrikaans • `am`—Amharic • `ar`—Arabic • `ar_AE`—Arabic (United Arab Emirates) • `ar_BH`—Arabic (Bahrain) • `ar_DZ`—Arabic (Algeria) • `ar_EG`—Arabic (Egypt) • `ar_IQ`—Arabic (Iraq) • `ar_JO`—Arabic (Jordan) • `ar_KW`—Arabic (Kuwait) • `ar_LB`—Arabic (Lebanon) • `ar_LY`—Arabic (Libya) • `ar_MA`—Arabic (Morocco) • `ar_OM`—Arabic (Oman) • `ar_QA`—Arabic (Qatar) • `ar_SA`—Arabic (Saudi Arabia) • `ar_SD`—Arabic (Sudan) • `ar_SY`—Arabic (Syria) • `ar_TN`—Arabic (Tunisia) • `ar_YE`—Arabic (Yemen) • `bg`—Bulgarian • `bn`—Bengali • `bs`—Bosnian • `ca`—Catalan • `cs`—Czech • `cy`—Welsh • `da`—Danish • `de`—German • `de_AT`—German (Austria) • `de_BE`—German (Belgium) • `de_CH`—German (Switzerland) • `de_LU`—German (Luxembourg) • `el`—Greek

Field	Details
	• en_AU—English (Australian) • en_CA—English (Canadian) • en_GB—English (UK) • en_HK—English (Hong Kong) • en_IE—English (Ireland) • en_IN—English (Indian) • en_MY—English (Malaysian) • en_NZ—English (New Zealand) • en_PH—English (Phillipines) • en_SG—English (Singapore) • en_US—English • en_ZA—English (South Africa) • eo—Esperanto (Pseudo) • es—Spanish • es_AR—Spanish (Argentina) • es_BO—Spanish (Bolivia) • es_CL—Spanish (Chile) • es_CO—Spanish (Colombia) • es_CR—Spanish (Costa Rica) • es_DO—Spanish (Dominican Republic) • es_EC—Spanish (Ecuador) • es_GT—Spanish (Guatemala) • es_HN—Spanish (Honduras) • es_MX—Spanish (Mexico) • es_NI—Spanish (Nicaragua) • es_PA—Spanish (Panama) • es_PE—Spanish (Peru) • es_PR—Spanish (Puerto Rico) • es_PY—Spanish (Paraguay) • es_SV—Spanish (El Salvador) • es_US—Spanish (United States) • es_UY—Spanish (Uruguay) • es_VE—Spanish (Venezuela) • et—Estonian • eu—Basque • fa—Farsi • fi—Finnish • fr—French

Field	Details
	• `fr_BE`—French (Belgium) • `fr_CA`—French (Canadian) • `fr_CH`—French (Switzerland) • `fr_LU`—French (Luxembourg) • `ga`—Irish • `gu`—Gujarati • `hi`—Hindi • `hr`—Croatian • `hu`—Hungarian • `hy`—Armenian • `in`—Indonesian • `is`—Icelandic • `it`—Italian • `it_CH`—Italian (Switzerland) • `iw`—Hebrew • `iw_EO`—Esperanto RTL (Pseudo) • `ja`—Japanese • `ka`—Georgian • `km`—Khmer • `kn`—Kannada • `ko`—Korean • `lb`—Luxembourgish • `lt`—Lithuanian • `lv`—Latvian • `mi`—Te reo • `mk`—Macedonian • `ml`—Malayalam • `mr`—Marathi • `ms`—Malay • `mt`—Maltese • `my`—Burmese • `nl_BE`—Dutch (Belgium) • `nl_NL`—Dutch • `no`—Norwegian • `pl`—Polish • `pt_BR`—Portuguese (Brazil) • `pt_PT`—Portuguese (European) • `rm`—Romansh

Field	Details
	• ro—Romanian • ro_MD—Romanian (Moldova) • ru—Russian • sh—Serbian (Latin) • sh_ME—Montenegrin • sk—Slovak • sl—Slovene • sq—Albanian • sr—Serbian (Cyrillic) • sv—Swedish • sw—Swahili • ta—Tamil • te—Telugu • th—Thai • tl—Tagalog • tr—Turkish • uk—Ukrainian • ur—Urdu • vi—Vietnamese • xh—Xhosa • zh_CN—Chinese (Simplified) • zh_HK—Chinese (Hong Kong) • zh_SG—Chinese (Singapore) • zh_TW—Chinese (Traditional) • zu—Zulu
MasterLabel	**Type** string **Properties** Filter, Group, Sort **Description** The label of the batch job.
ProcessGroup	**Type** string **Properties** Filter, Group, Sort

Field	Details
	Description The group or team that's using the batch job. This field is only applicable to Batch Management jobs.
`Status`	**Type** picklist **Properties** Defaulted on create, Filter, Group, Restricted picklist, Sort **Description** The status of the batch job. Possible values are: • `Active` • `Inactive`
`Type`	**Type** picklist **Properties** Defaulted on create, Filter, Group, Restricted picklist, Sort **Description** The type of batch job. Possible values are: • `Calc`—Data Processing Engine • `DecisionTableRefresh` • `Flow`

Common Platform Event

Batch Management jobs and Data Processing Engine definitions are run using invocable actions in Flows. Use the BatchJobStatusChanged event to notify subscribers after a Batch Management job or a Data Processing Engine definition is processed in a flow.

BatchJobStatusChangedEvent
Notifies subscribers of when a batch job is completed in a flow. This object is available in API version 51.0 and later.

EDITIONS

Available in: Lightning Experience

Available in: Data Processing Engine is available with Enterprise, Unlimited, and Performance Editions with the Financial Services Cloud, Loyalty Management, Manufacturing Cloud, Rebate Management, Accounting Subledger, or Provider Search in Health Cloud.

Available in: Batch Management is available with Enterprise, Unlimited, and Performance Editions with the Loyalty Management, Manufacturing Cloud, Rebate Management, or Accounting Subledger

BatchJobStatusChangedEvent

Notifies subscribers of when a batch job is completed in a flow. This object is available in API version 51.0 and later.

Supported Calls

`describeSObjects()`

Supported Subscribers

Subscriber	Supported?
Flows	✔

Fields

Field	Details
`BatchJob`	**Type** string **Description** The unique identifier of the batch job.
`BatchJobDefinition`	**Type** string **Properties** Nillable **Description** The unique identifier of the batch job's definition.
`EndDateTime`	**Type** dateTime

Field	Details
	Properties Nillable **Description** The timestamp for when the batch job execution is complete.
`EventUuid`	**Type** string **Properties** Nillable **Description** A universally unique identifier (UUID) that identifies a platform event message. This field is available in API version 52.0 and later.
`ReplayId`	**Type** string **Properties** Nillable **Description** Represents an ID value that is populated by the system and refers to the position of the event in the event stream. Replay ID values aren't guaranteed to be contiguous for consecutive events. A subscriber can store a replay ID value and use it on resubscription to retrieve missed events that are within the retention window.
`StartDateTime`	**Type** dateTime **Properties** Nillable **Description** The timestamp for when the batch job execution is started.
`Status`	**Type** picklist **Properties** Restricted picklist **Description** The status of the batch job. Possible values are: • `Canceled`—Canceled • `Failure` • `Success`

Common Business APIs

Common Business APIs are RESTful APIs that are sometimes available as Apex classes and methods.

REST Reference
You can access Common Business APIs using REST endpoints. These REST APIs follow similar conventions as Connect REST APIs.

REST Reference

You can access Common Business APIs using REST endpoints. These REST APIs follow similar conventions as Connect REST APIs.

To understand the architecture, authentication, rate limits, and how the requests and responses work, see Connect REST API Developer Guide.

Resources
Here's a list of Common Business API resources.

Response Bodies
Here's a list of Common Business API response bodies.

SEE ALSO:
Connect REST API Developer Guide

Resources

Here's a list of Common Business API resources.

Batch Job Cancel
Cancel a batch job of type data processing engine (calc job) and batch management. A batch job with only the status Submitted or In Progress can be canceled.

Batch Job Cancel

Cancel a batch job of type data processing engine (calc job) and batch management. A batch job with only the status Submitted or In Progress can be canceled.

Special Access Rules

To use this resource, the following permissions are required:

- Your org must have the Batch Management and Data Processing Engine licenses
- Users in your org require System Administration profile

Resource

```
/connect/batch-job/batchJobId/cancel-job
```

Resource example

```
/connect/batch-job/0mdxx00000000fxAAA/cancel-job
```

Available version
52.0

Requires Chatter
No

HTTP methods
POST

Note: POST doesn't take request parameters or a request body.

Response body for POST
Returns HTTP 201 on success.

See Batch Job Cancel Output for HTTP code descriptions that are unique to this resource in case of failure of the batch job cancel request.

Response Bodies

Here's a list of Common Business API response bodies.

Batch Job Cancel Output
Output representation of the batch job cancel request.

Batch Job Cancel Output

Output representation of the batch job cancel request.

Property Name	Type	Description	Filter Group and Version	Available Version
`message`	String	Details about why the batch job cancel request failed.	Small, 52.0	52.0

This table lists HTTP response code descriptions that are unique to this resource.

HTTP Response Code	Error Code	Description
400	`INVALID_STATUS`	We can't cancel the batch job that doesn't have an active run. Specify the ID of a batch job with an active run and try again.
400	`INVALID_STATUS`	We can't cancel the batch job that is already canceled or completed. Specify the ID of a valid batch job with the status InProgress or Submitted and try again.
400	`DELETE_FAILED`	We can't cancel the batch job of which the results are already being written back.
400	`DELETE_FAILED`	We can't cancel the batch job because of an error in processing your org's data. Run the Data Processing Engine definition and try again.

HTTP Response Code	Error Code	Description
403	`FORBIDDEN`	You don't have the permission to cancel a batch job. Ask your Salesforce admin for help.
404	`RESOURCE_NOT_FOUND`	Specify the ID of a valid batch job and try again.
500	`INTERNAL_SERVER_ERROR`	Something went wrong when we tried to cancel the batch job. Try again or ask your Salesforce admin for help.

Data Processing Engine

Transform data that's available in your Salesforce org and write back the transformation results as new or updated records. You can transform the data for standard and custom objects. Data Processing Engine consists of a Tooling API object, a standard object, a Metadata API, and an invocable action. You can use these to view, create, edit, and run Data Processing Engine definitions.

Data Processing Engine Tooling API Objects

Data Processing Engine consists of one Tooling API object, BatchCalcJobDefinition. Use this object to create and edit a Data Processing Engine definition.

Data Processing Engine Standard Object

Data Processing Engine contains one standard object, BatchCalcJobDefinitionView. Use this object to view all the Data Processing Engine definitions available in your Salesforce org, including file-based definitions.

Data Processing Engine Metadata API

Use a Metadata API to create, update, and activate Data Processing Engine definitions.

Data Processing Engine Invocable Actions

Run an active Data Processing Engine definition. For more information on custom invocable actions, see **REST API Developer Guide** and **Actions Developer Guide**.

EDITIONS

Available in: Lightning Experience

Available in: Data Processing Engine is available with Enterprise, Unlimited, and Performance Editions with the Financial Services Cloud, Loyalty Management, Manufacturing Cloud, Rebate Management, Accounting Subledger, or Provider Search in Health Cloud.

Data Processing Engine Tooling API Objects

Data Processing Engine consists of one Tooling API object, BatchCalcJobDefinition. Use this object to create and edit a Data Processing Engine definition.

Tooling API objects let you interact with metadata for declarative development. For example, you can create your own version of Setup.

BatchCalcJobDefinition
Represents a Data Processing Engine (DPE) definition. This object is available in API version 51.0 and later.

EDITIONS

Available in: Lightning Experience

Available in: Data Processing Engine is available with Enterprise, Unlimited, and Performance Editions with the Financial Services Cloud, Loyalty Management, Manufacturing Cloud, Rebate Management, Accounting Subledger, or Provider Search in Health Cloud.

BatchCalcJobDefinition

Represents a Data Processing Engine (DPE) definition. This object is available in API version 51.0 and later.

Note: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Supported SOAP API Calls

`describeSObjects()`, `query()`, `retrieve()`

Supported REST API Methods

`DELETE, GET, HEAD, PATCH, POST, Query`

Fields

Field	Details
`BatchJobDefinitionId`	**Type** reference **Properties** Filter, Group, Nillable, Sort **Description** The unique identifier of the associated batch job definition. This is a relationship field. **Relationship Name** BatchJobDefinition **Relationship Type** Lookup **Refers To** BatchJobDefinition

Field	Details
`DefinitionRunMode`	**Type** picklist **Properties** Create, Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort, Update **Description** Possible values are: • `Batch` • `OnDemand`—On Demand The default value is `Batch`.
`DeveloperName`	**Type** string **Properties** Filter, Group, Sort **Description** The developer name of the Data Processing Engine definition.
`ExecutionPlatformType`	**Type** Picklist **Properties** Filter, Group, Restricted picklist, Sort, Not-Nillable **Description** Specifies the platform that's used to run the Data Processing Engine definition. Possible values: • `CRMA`—CrmAnalytics • `CDP`— DataCloudForMarketing • `CORE`—This value is reserved for internal use.
`FullName`	**Type** string **Properties** Create, Group, Nillable **Description** The name of the Data Processing Engine definition. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.
`IsTemplate`	**Type** boolean

Field	Details
	Properties Defaulted on create, Filter, Group, Sort **Description** Indicates whether this is a template Data Processing Engine definition.
`Language`	**Type** picklist **Properties** Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort **Description** The language in which this Data Processing Engine definition is created. Possible values are: • `da`—Danish • `de`—German • `en_US`—English • `es`—Spanish • `es_MX`—Spanish (Mexico) • `fi`—Finnish • `fr`—French • `it`—Italian • `ja`—Japanese • `ko`—Korean • `nl_NL`—Dutch • `no`—Norwegian • `pt_BR`—Portuguese (Brazil) • `ru`—Russian • `sv`—Swedish • `th`—Thai • `zh_CN`—Chinese (Simplified) • `zh_TW`—Chinese (Traditional)
`ManageableState`	**Type** ManageableState enumerated list **Properties** Filter, Group, Nillable, Restricted picklist, Sort **Description** Indicates the manageable state of the specified component that is contained in a package: • `beta` • `deleted`

Field	Details
	• `deprecated` • `deprecatedEditable` • `installed` • `installedEditable` • `released` • `unmanaged`
`MasterLabel`	**Type** string **Properties** Filter, Group, Sort **Description** Label of the Data Processing Engine definition.
`Metadata`	**Type** complexvalue **Properties** Create, Nillable, Update **Description** Data Processing Engine definition's metadata. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.
`NamespacePrefix`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The namespace prefix associated with this object. Each Developer Edition organization that creates a managed package has a unique namespace prefix. Limit: 15 characters. You can refer to a component in a managed package by using the ***`namespacePrefix__componentName`*** notation. The namespace prefix can have one of the following values: • In Developer Edition organizations, the namespace prefix is set to the namespace prefix of the organization for all objects that support it. There's an exception if an object is in an installed managed package. In that case, the object has the namespace prefix of the installed managed package. This field's value is the namespace prefix of the Developer Edition organization of the package developer. • In organizations that aren't Developer Edition organizations, `NamespacePrefix` is only set for objects that are part of an installed managed package. There's no namespace prefix for all other objects.

Field	Details
`ProcessType`	**Type** picklist **Properties** Filter, Group, Restricted picklist, Sort **Description** Required. The process type for which the Data Processing Engine definition is created. Possible values are: • `AccountingPeriodClosure`—Legal Entity Accounting Period Closure • `AccountingSubledger`—This value is reserved for internal use. • `ActionableList` • `AdvancedAccountForecast` • `BenefitManagement`—Available in API version 61.0 and later. • `BillingSchedulesforInvoiceGeneration`—Billing Schedules for Invoice Generation • `ChannelInventoryManagement`—Available in API version 63.0 and later. • `CriteriaBsdSearchAndFilter - Criteria-Based Search And Filter` • `CSVCoreUpload`—This value is reserved for internal use. • `DataProcessingEngine - Standard` • `DecisionMatrixDataUpload` - This value is available only if you have Business Rules Engine enabled. • `Decisiontable`—Decision table activation • `EmployeeService`—Available in API version 63.0 and later. • `FinancialSummaryRollup`—Available in API version 62.0 and later. • `FSCHierarchyRollUp` • `Fundraising`—Available in API version 64.0 and later. • `FundraisingRollups`—Available in API version 63.0 and later. • `InvoiceGeneration`—Available in API version 62.0 and later. • `LegalEntityAccountingPeriodClosureAdvanced`—Available in API version 63.0 and later. • `LifeSciencesCommercialTerritoryAlignment`—Available in API version 63.0 and later. • `LifeSciencesCustomerEngagement`—Available in API version 64.0 and later. • `Loyalty` • `LoyaltyPartnerManagement` • `LoyaltyPointsAggregation` • `NetZero` • `NextGenForecasting`—Available in API version 62.0 and later. • `PatientServicesProgram`—Available in API version 64.0 and later.

Field	Details
	• `PnmRosterFileUpload`—Available in API version 62.0 and later. • `PriceProtection`—Available in API version 62.0 and later. • `ProductCatalogManagement`—Available in API version 62.0 and later. • `ProgramBasedBusiness` • `ProgramManagementRollups` • `ProviderSearch`—This value is reserved for internal use. • `Rebates` • `Recruitment`—Available in API version 62.0 and later. • `RevenueTransactionManagement`—Available in API version 63.0 and later. • `SalesAgreement`—Available in API version 63.0 and later. • `TestAtomicWritebackScale`—Available in API version 64.0 and later. • `TestProcessType`—Available in API version 62.0 and later. • `UsageManagement`—Available in API version 62.0 and later. When Data Processing Engine is enabled for a Salesforce org, the default value is 'Standard'. Other process types may be available to you depending on your industry solution and permission sets.

Data Processing Engine Standard Object

Data Processing Engine contains one standard object, BatchCalcJobDefinitionView. Use this object to view all the Data Processing Engine definitions available in your Salesforce org, including file-based definitions.

BatchCalcJobDefinitionView
Represents the details of a Data Processing Engine definition. The definition can also be a file-based definition that is available in your Salesforce org. This object is available in API version 51.0 and later.

EDITIONS

Available in: Lightning Experience

Available in: Data Processing Engine is available with Enterprise, Unlimited, and Performance Editions with the Financial Services Cloud, Loyalty Management, Manufacturing Cloud, Rebate Management, Accounting Subledger, or Provider Search in Health Cloud.

BatchCalcJobDefinitionView

Represents the details of a Data Processing Engine definition. The definition can also be a file-based definition that is available in your Salesforce org. This object is available in API version 51.0 and later.

Note: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Supported Calls

`describeSObjects()`, `query()`

Fields

Field	Details
`DataSpaceApiName`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** Stores the data space API name from Data Cloud.
`Description`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The description of a Data Processing Engine definition.
`DefinitionRunMode`	**Type** picklist **Properties** Create, Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort, Update **Description** Possible values are: • `Batch` • `OnDemand`—On Demand The default value is `Batch`.
`DurableId`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The unique identifier for the field. Always retrieve this value before using it, as the value isn't guaranteed to stay the same from one release to the next. Simplify queries by using this field instead of making multiple queries.
`ExecutionPlatformType`	**Type** Picklist

Field	Details
	Properties **Filter, Group, Restricted picklist, Sort, Not-Nillable** **Description** Specifies the platform that's used to run the Data Processing Engine definition. Possible values: • CRMA: CrmAnalytics • CDP: DataCloudForMarketing
`InstalledPackageName`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The name of the package used to add the definition to the org.
`IsActive`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** Indicates whether the definition is active.
`IsTemplate`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** Indicates whether the definition is a template. You can make a copy of a template definition and update it based on your requirements.
`LastModifiedBy`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The unique identifier of the user who modified the definition last.
`ManageableState`	**Type** picklist **Properties** Filter, Group, Nillable, Restricted picklist, Sort

Field	Details
	Description Indicates the manageable state of the definition if it's contained in a package. Possible values are: • `beta`—Managed-Beta • `deleted`—Managed-Proposed-Deleted • `deprecated`—Managed-Proposed-Deprecated • `deprecatedEditable`—SecondGen-Installed-Deprecated • `installed`—Managed-Installed • `installedEditable`—SecondGen-Installed-Editable • `released`—Managed-Released • `unmanaged`—Unmanaged
`MasterLabel`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The label of the Data Processing Engine definition.
`Name`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The name of the Data Processing Engine definition.
`NamespacePrefix`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The namespace prefix associated with this object. Each Developer Edition organization that creates a managed package has a unique namespace prefix. Limit: 15 characters. You can refer to a component in a managed package by using the namespacePrefix__componentName notation. The namespace prefix can have one of the following values: • In Developer Edition organizations, the namespace prefix is set to the namespace prefix of the organization for all objects that support it. There's an exception if an object is in an installed managed package. In that case, the object has the namespace prefix of the installed managed package. This field's value is the namespace prefix of the Developer Edition organization of the package developer.

Field	Details
	• In organizations that aren't Developer Edition organizations, NamespacePrefix is only set for objects that are part of an installed managed package. There's no namespace prefix for all other objects.
`ProcessType`	**Type** picklist **Properties** Filter, Group, Nillable, Restricted picklist, Sort **Description** The process type of the definition. Possible values are: • `AccountingPeriodClosure`—Legal Entity Accounting Period Closure—Available in API version 62.0 and later. • `ActionableList` • `AdvancedAccountForecast` • `BenefitManagement`—Available in API version 61.0 and later. • `BillingSchedulesforInvoiceGeneration`—Billing Schedules for Invoice Generation—Available in API version 62.0 and later. • `ChannelInventoryManagement`—Available in API version 63.0 and later. • `CriteriaBsdSearchAndFilter - Criteria-Based Search And Filter` • `DataProcessingEngine - Standard` • `DecisionMatrixDataUpload`-This value is available only if you have Business Rules Engine enabled. • `Decisiontable`—Decision table activation—Available in API version 62.0 and later. • `EmployeeService`—Available in API version 63.0 and later. • `FinancialSummaryRollup`—Available in API version 63.0 and later. • `FSCHierarchyRollUp` • `FundraisingRollups`—Available in API version 63.0 and later. • `InvoiceGeneration`—Available in API version 62.0 and later. • `LegalEntityAccountingPeriodClosureAdvanced`—Available in API version 63.0 and later. • `LifeSciencesCommercialTerritoryAlignment`—Available in API version 63.0 and later. • `Loyalty` • `LoyaltyPartnerManagement` • `LoyaltyPointsAggregation` • `NetZero` • `NextGenForecasting`—Available in API version 62.0 and later. • `PnmRosterFileUpload`—Available in API version 62.0 and later.

Field	Details
	• `PriceProtection`—Available in API version 62.0 and later. • `ProductCatalogManagement`—Available in API version 63.0 and later. • `ProgramBasedBusiness` • `ProgramManagementRollups` • `Rebates` • `RevenueTransactionManagement`—Available in API version 63.0 and later. • `AccountingSubledger`—This value is reserved for internal use. • `ProviderSearch`—This value is reserved for internal use. • `Recruitment`—Available in API version 62.0 and later. • `SalesAgreement`—Available in API version 63.0 and later. • `TestProcessType`—Available in API version 62.0 and later. • `UsageManagement`—Available in API version 62.0 and later. When Data Processing Engine is enabled for a Salesforce org, the default value is 'Standard'. Other process types may be available to you depending on your industry solution and permission sets.

Data Processing Engine Metadata API

Use a Metadata API to create, update, and activate Data Processing Engine definitions.

BatchCalcJobDefinition
Represents a Data Processing Engine definition.

EDITIONS

Available in: Lightning Experience

Available in: Data Processing Engine is available with Enterprise, Unlimited, and Performance Editions with the Financial Services Cloud, Loyalty Management, Manufacturing Cloud, Rebate Management, Accounting Subledger, or Provider Search in Health Cloud.

BatchCalcJobDefinition

Represents a Data Processing Engine definition.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

BatchCalcJobDefinition components have the suffix `.batchCalcJobDefinition` and are stored in the `batchCalcJobDefinitions` folder.

Version

BatchCalcJobDefinition components are available in API version 51.0 and later.

Special Access Rules

To use this metadata type, one of these licenses is required:

- Loyalty Management
- Financial Services Cloud
- Rebate Management
- Manufacturing Cloud
- Net Zero Cloud

Fields

Field Name	Field Type	Description
`aggregates`	BatchCalcJobAggregate[]	Collection of aggregate nodes in a data processing engine.
`appends`	BatchCalcJobUnion[]	Collection of append nodes in a data processing engine.
`atomicWritebacks`	BatchCalcJobAtomicWriteBack[]	Collection of composite writeback nodes in a data processing engine definition. Available in API version 62.0 and later.
`customNodes`	BatchCalcJobCustomNode[]	Collection of custom nodes in a data processing engine. Available in API version 57.0 and later.
`dataSpaceApiName`	string	Stores the Data Space API Name from Data Cloud. Available in API version 60.0 and later.
`datasources`	BatchCalcJobDatasource[]	Collection of data source nodes in a data processing engine.
`definitionRunMode`	BatchCalcJobRunModeEnum (enumeration of type string)	Specifies the execution mode in a data processing engine. Valid values are: • `Batch` • `OnDemand`
`description`	string	Description of a data processing engine definition.
`executionPlatformType`	ExecutionPlatformTypeEnum (enumeration of type string)	The platform that's used to run the Data Processing Engine definition. Valid values are: • `CRMA` • `CDP` • `CORE` Available in API version 59.0 and later.
`filters`	BatchCalcJobFilter[]	Collection of filter nodes in a data processing engine. definition.
`forecasts`	BatchCalcJobForecast[]	Collection of forecast nodes in a data processing engine. definition. Available in API version 58.0 and later.
`hierarchyPaths`	BatchCalcJobHierarchyPath[]	Collection of hierarchy path nodes in a data processing engine definition.
`isTemplate`	boolean	Indicates whether it's a template data processing engine definition.

Field Name	Field Type	Description
`joins`	BatchCalcJobSourceJoin[]	Collection of join nodes in a data processing engine.
`label`	string	The label of a data processing engine definition.
`parameters`	BatchCalcJobParameter[]	Collection of input variables in a data processing engine.
`processType`	BatchCalcProcessType (enumeration of type string)	The process type of a data processing engine. These process types may be available to you depending on your industry solution and permission sets. Valid values are: • `AccountingPeriodClosure` • `AccountingSubledger`—This value is reserved for internal use. • `ActionableList` • `AdvancedAccountForecast` • `BenefitManagement` • `BillingSchedulesforInvoiceGeneration` • `CDPEnrichment` • `ChannelInventoryManagement`—Available in API version 63.0 and later. • `CriteriaBsdSearchAndFilter` • `DataProcessingEngine` • `DecisionMatrixDataUpload` • `Decisiontable` • `Education` • `EmployeeService`—Available in API version 63.0 and later. • `FinancialSummaryRollup` • `FSCHierarchyRollUp` • `Fundraising`—Available in API version 64.0 and later. • `FundraisingRollups`—Available in API version 63.0 and later. • `InvoiceGeneration` • `Loyalty` • `LegalEntityAccountingPeriodClosureAdvanced`—Available in API version 63.0 and later. • `LifeSciencbatchcalesCommercialTerritoryAlignment`—Available in API version 63.0 and later. • `LifeSciencesCustomerEngagement`—Available in API version 64.0 and later. • `LoyaltyPartnerManagement` • `LoyaltyPointsAggregation` • `NextGenForecasting` • `NetZero`

Field Name	Field Type	Description
		• `PatientServicesProgram`—Available in API version 64.0 and later. • `PnmRosterFileUpload` • `PriceProtection` • `ProductCatalogManagement` • `ProgramBasedBusiness` • `ProviderSearch`—This value is reserved for internal use. • `Rebates` • `Recruitment` • `RevenueTransactionManagement`—Available in API version 63.0 and later. • `SalesAgreement`—Available in API version 63.0 and later. • `TestAtomicWritebackScale`—Available in API version 64.0 and later. • `TestProcessType` • `UsageManagement`
`status`	BatchJobDefinition Status (enumeration of type string)	Status of a data processing engine definition. Valid values are: • `Active` • `Inactive`
`transforms`	BatchCalcJobTransform[]	Collection of data transformation nodes in a data processing engine.
`writebacks`	BatchCalcJobWriteback Object[]	Collection of writeback objects in which the results of the data processing engine are written back.

BatchCalcJobAggregate

Represents a collection of fields relating to an aggregate node in a data processing engine.

Fields

Field Name	Field Type	Description
`description`	string	Description of an aggregate node.
`fields`	BatchCalcJob AggregateField[]	Required. Collection of aggregation fields.
`groupBy`	string[]	Required. Collections of fields used to group data in an aggregate node.
`label`	string	Required. Label of an aggregate node.
`name`	string	Required. Name of an aggregate node.
`sourceName`	string	Required. Name of the source node.

BatchCalcJobAggregateField

Represents a collection of fields relating to an aggregation field in an aggregate node of a data processing engine.

Fields

Field Name	Field Type	Description
`aggregateFunction`	BatchCalcJobAggregateFunction (enumeration of type string)	Required. Function used for aggregation. Valid values are: • `Unique`—A count of unique values. • `Sum`—The sum of all values. • `Max`—The largest value. • `Min`—The smallest value. • `Avg`—The average value, calculated as the mean. • `Std`—The standard deviation. • `Stdp`—A standard deviation with population variance. • `Var`—The variance. • `VarP`—The variance with population. • `Count`—The total count of values.
`alias`	string	Required. Name that subsequent nodes within the data processing engine use to refer to the aggregate field.
`sourceFieldName`	string	Required. Source node field on which the aggregate is calculated.

BatchCalcJobAtomicWriteback

Represents a node in a DPE definition that stores the details about the relationship between the writeback nodes and the composite writeback operations between the nodes.

Field Name	Field Type	Description
`description`	string	Description of the composite writeback object.
`label`	string	Required. Name of the composite writeback object.
`name`	string	Required. API name of the composite writeback object.
`writebackObject Relationships`	BatchCalcJobAtomicWritebackRelationship[]	Specifies the relationship between the writeback objects that are involved in the writeback operation.
`writebackSequence`	int	Sequence in which the data processing engine executes the composite write back node.

BatchCalcJobAtomicWritebackRelationship

Represents the relationships between the writeback objects that are involved in a composite writeback operation. It captures the relationships between these objects and the sequence in which they should be processed.

Field Name	Field Type	Description
`childWritebackObjectField`	string	Field name that's associated with the child writeback object in a composite writeback relationship. Available in API version 63.0 and later.
`childWritebackObjectName`	string	Name of the child writeback object that's associated with the writeback relationship.
`parentWritebackObjectField`	string	Field name that's associated with the parent writeback object in a composite writeback relationship. Available in API version 63.0 and later.
`parentWritebackObjectName`	string	Required. Name of the parent writeback object that's associated with the writeback relationship.
`relationshipName`	string	Describes the relationship between the child and parent writeback objects in a composite writeback node. Available in API version 64.0 and later.
`sequenceNumber`	int	Sequence number of the writeback node that's associated with its parent node in the relationship.

BatchCalcJobCustomNode

Represents a collection of custom nodes in a data processing engine. Use a custom node to add a custom action.

Fields

Field Name	Field Type	Description
`description`	string	Description of a custom node.
`extensionName`	string	Required. Name of an extension node.
`extensionNamespace`	string	Required. Namespace of an extension node.
`label`	string	Required. Label of a custom node.
`name`	string	Required. Name of a custom node.
`parameters`	BatchCalcJobCustomNodeParameter[]	The field mappings of an extension node.
`sources`	string[]	Sources of an extension node.

BatchCalcJobCustomNodeParameter

Represents the field mappings of an extension node.

Fields

Field Name	Field Type	Description
`name`	string	Required. Name of a parameter.
`value`	string	Required. Value of a parameter.

BatchCalcJobDatasource

Represents a collection of fields relating to a data source node in a data processing engine.

Fields

Field Name	Field Type	Description
`CSVDelimiter`	BatchCalcJobCSVDelimiter (enumeration of type string)	Specifies the field separator to read fields from a CSV file record. Possible values are: • `COMMA` • `BACKQUOTE` • `CARET` • `PIPE` • `SEMICOLON` • `TAB` The default value is `COMMA`. The same delimiter value used for the CSV file can't be used within any of the column values in the file. If you mistakenly use the same delimiter value in column values, it can cause data parsing issues.
`description`	string	Description of a data source node.
`fields`	BatchCalcJob DatasourceField[]	Required. Collection of data source fields.
`fileIdentifier`	string	Specifies the source of the file or file storage system.
`filePath`	string	The file path for the specified file.
`fileSource`	BatchCalcJobFileSource (enumeration of type string)	Specifies the source of the file or file storage system. Possible value is: • `ContentManagement`
`label`	string	Required. Label of a data source node.
`name`	string	Required. Name of a data source node.
`sourceName`	string	Required. Name of a standard or custom object from which the data source node extracts data.

Field Name	Field Type	Description
`type`	BatchCalcJobDataSource Type (enumeration of type string)	Required. Type of object for the source object field. Supported values are: • `Analytics` • `CalculatedInsightsObject` • `CRMObject` • `CSV` • `DataModelObject` • `StandardObject` .

BatchCalcJobDatasourceField

Represents a collection of fields relating to a source object field that are selected in the data source node of a data processing engine.

Fields

Field Name	Field Type	Description
`alias`	string	Name that subsequent nodes within the data processing engine use to refer to the data source field. Required when the field name is lookup.
`dataType`	BatchCalcJobDataType (enumeration of type string)	Specifies the data type of the input field when using a CSV file as a data source. Possible values are: • `Text` • `Numeric` • `Date` • `DateTime` • `MultiValue`
`isPrimaryKey`	boolean	Indicates whether a column name is the primary key (`true`) or not (`false`) for the Data Cloud CSV file.
`name`	string	Required. Name of the field. Can be either of the following: • Name of the source field selected in the associated data source object. • Name from a nested lookup object with three child levels.

BatchCalcJobFilter

Represents a collection of fields relating to a filter node in a data processing engine.

Fields

Field Name	Field Type	Description
`criteria`	BatchCalcJobFilterCriteria[]	Collection of filter criteria in a filter node. The field is required when `isDynamicFilter` is set to `False`.
`description`	string	Description of the batch calculation job filter.
`filterCondition`	string	Logic that is specified to apply the filter conditions. The field is required when `isDynamicFilter` is set to `False`.
`filterParameterName`	string	Name of the parameter of type filter.
`isDynamicFilter`	boolean	Indicates whether the filter criteria is dynamic. If value is set to `True`, filter criteria is passed in runtime with `filterParameterName`.
`label`	string	Required. Label of the filter node.
`name`	string	Required. Name of the filter node.
`sourceName`	string	Required. Name of the source node.

BatchCalcJobForecast

Represents a collection of fields relating to a forecast node in a data processing engine. Available in API version 58.0 and later.

Fields

Field Name	Field Type	Description
accuracyPercent	BatchCalcJobFrcstAccuracy (enumeration of type string)	The interval percentage to account for errors in forecasts. Possible values are: • `Eighty` • `NinetyFive` • `None` The default value is `None`.
aggregateFields	BtchCalcJobFrcstAggrFld[]	The list of fields to forecast.
dateFieldName	string	Required. The date field from the source node used to forecast values for the specified forecast length.

Field Name	Field Type	Description
description	string	The description of the forecast node.
forecastModelType	BatchCalcJobFrcstModel (enumeration of type string)	The model used to forecast data. Possible values are: • `Additive` • `Auto` • `Multiplicative` The default value is `Auto`.
forecastPeriodCount	int	The number of time periods to generate forecast data. For example, if you select Year-Month as the forecast period type, and 4 as the forecast period count, the forecast results are generated for the next 4 months. The minimum and the default count is 1, and the maximum is 100.
forecastPeriodType	BatchCalcJobFrcstPeriodType (enumeration of type string)	Required. The type of forecast period to group date field values in the forecast results. Possible values are: • `FiscalYear` • `FiscalYearMonth` • `FiscalYearQuarter` • `FiscalYearWeek` • `Year` • `YearMonth` • `YearMonthDay` • `YearQuarter` • `YearWeek`
groupFields	BatchCalcJobFrcstGrpFld[]	The source fields for grouping the data to be

Field Name	Field Type	Description
		processed by the forecast node.
`label`	string	Required. The name of the forecast node in the UI.
`name`	string	Required. A unique name for the forecast node.
[illegible]	string	Required. The start date of the forecast period.
[illegible]	BatchCalcJobFrcstSeasonality (enumeration of type string)	Represents the periodic fluctuations that occur around the same time every year. Possible values are: • `Two` • `Three` • `Four` • `Five` • `Six` • `Seven` • `Eight` • `Nine` • `Ten` • `Eleven` • `Twelve` • `Thirteen` • `Fourteen` • `Fifteen` • `Sixteen` • `Seventeen` • `Eighteen` • `Nineteen` • `Twenty` • `TwentyOne`

Field Name	Field Type	Description
		• `TwentyTwo` • `TwentyThree` • `TwentyFour` • `Auto` • `None` The default value is `None`.
[illegible]	boolean	Indicates whether to ignore the last period in the source node when it has incomplete data (`true`) or not (`false`). The default value is `false`.
sourceName	string	Required. The name of the source node. A source can be any node other than the datasink and register node.

BtchCalcJobFrcstAggrFld

Represents a list of fields to forecast in a forecast node.

Field Name	Field Type	Description
`aggregateFunction`	BatchCalcJobAggregateFunction (enumeration of type string)	Required. The function of the aggregate field. Possible values are: • `Avg` • `Count` • `Max` • `Min` • `Std` • `StdP` • `Sum` • `Unique` • `Var`

Field Name	Field Type	Description
		• `VarP`
`aggregationResultLabel`	string	Required. The name of the aggregation result generated from the aggregation function that's applied to the source node field.
`fieldName`	string	Required. The name of the source field.

BatchCalcJobFrcstGrpFld

Represents source fields for grouping the data to be processed by the forecast node.

Field Name	Field Type	Description
`fieldName`	string	Required. The name of the source field to group the data to be processed by the forecast node.
`groupBy`	string	A comma-separated list of values to group data by. Required when the source field type is Date or DateTime. Possible values are: • `Second` • `Second Epoch` • `Minute` • `Hour` • `Day` • `Day Epoch` • `Week` • • `Month` • `Quarter` • `Year`

BatchCalcJobHierarchyPath

Represents a collection of hierarchy path nodes in a data processing engine definition.

Fields

Field Name	Field Type	Description
`description`	string	Description of the hierarchy path node.
`hierarchyFieldName`	string	Required. Field name that contains the hierarchy path.
`isSelfFieldValueIncluded`	boolean	Indicates whether the self value is included in the calculated hierarchy path (`True`) or not (`False`).
`label`	string	Required. Label of the hierarchy path node.
`name`	string	Required. Name of the hierarchy path node.
`parentFieldName`	string	Required. Parent field name to calculate hierarchy path.
`selfFieldName`	string	Required. Self field name to calculate hierarchy path.
`sourceName`	string	Required. Name of the source node.

BatchCalcJobFilterCriteria

Represents a collection of fields relating to a filter condition in a filter node in a data processing engine.

Fields

Field Name	Field Type	Description
`inputVariable`	string	Name of the input variable used as a filter.
`operator`	BatchCalcJobFilter Operator (enumeration of type string)	Required. Operator that is specified in the filter condition. Valid values are: • `Equals` • `NotEquals` • `GreaterThan` • `GreaterThanOrEqual` • `LessThan` • `LessThanOrEqual` • `StartsWith` • `EndsWith` • `Contains` • `DoesNotContain` • `IsNull` • `IsNotNull` • `In` • `NotIn`

Field Name	Field Type	Description
`sequence`	integer	Required. Sequence number used to refer the criteria in a filter node.
`sourceFieldName`	string	Required. Name of the field from the source node to apply the filter.
`value`	string	Value used to filter data from the source node.

BatchCalcJobParameter

Represents a collection of fields relating to an input variable in a data processing engine.

Fields

Field Name	Field Type	Description
`dataType`	BatchCalcJobParameterDataType (enumeration of type string)	Required. Data type of the parameter. Valid values are: • `Date` • `DateTime` • `Expression` • `FileIdentifier` • `Filter` • `Numeric` • `Text`
`defaultValue`	string	Default value of the parameter.
`description`	string	Description of the batch calculation job parameter.
`isMultiValue`	boolean	Indicates whether the parameter has different values (`True`) or not (`False`). This field is supported only for the `Text` data type.
`label`	string	Required. Label of the batch calculation job parameter.
`name`	string	Required. Name of the batch calculation job parameter.

BatchCalcJobSourceJoin

Represents a collection of fields relating to a join node in a data processing engine.

Fields

Field Name	Field Type	Description
`description`	string	Description of the join node.
`fields`	BatchCalcJobJoinResultField[]	Collection of fields in a join node.

Field Name	Field Type	Description
`joinKeys`	BatchCalcJobJoinKey[]	Collection of mapping of fields from the primary source node and the second source node in a join node.
`label`	string	Required. Label of the join node.
`name`	string	Required. Name of the join node.
`primarySourceName`	string	Required. Name associated with the node as the primary source node.
`secondarySourceName`	string	Required. Name associated with the node as the secondary source node.
`type`	BatchCalcJobSourceJoinType (enumeration of type string)	Required. Type of join specified between the primary source node and secondary source node. Valid values are: • `LeftOuter` • `RightOuter` • `Inner` • `Outer` • `Lookup`

BatchCalcJobJoinKey

Represents a collection of fields relating to a mapping of fields from the first source node and second source node in a join node of a data processing engine.

Fields

Field Name	Field Type	Description
`primarySourceFieldName`	string	Required. Mapped field name of the primary source node.
`secondarySourceFieldName`	string	Required. Mapped field name of the secondary source node.

BatchCalcJobJoinResultField

Represents a collection of fields relating to a set of resultant fields in a join node of a data processing engine.

Fields

Field Name	Field Type	Description
`alias`	string	Required. Name that subsequent nodes within the data processing engine definition use to refer to the resultant field.
`sourceFieldName`	string	Required. Name of field from the primary or secondary data source.
`sourceName`	string	Required. Source node of the primary or secondary data source.

BatchCalcJobTransform

Represents a collection of fields relating to a data transformation in a data processing engine.

Fields

Field Name	Field Type	Description
`description`	string	The description of the batch calculation job transform.
`droppedFields`	BatchCalcJobTransformDroppedField[]	The collection of dropped fields in a data transformation. Available when the transformation type is `Slice`.
`expressionFields`	BatchCalcJobTransformAddedField[]	The collection of formula fields in a data transformation. Available when the transformation type is `Expression`.
`label`	string	Required. The label of the batch calculation job transform.
`name`	string	Required. The name of the batch calculation job transform.
`orderBy`	BatchCalcJobOrderByField on page 137[]	A collection of fields that's used to sort the records within each partition group.
`partitionBy`	string[]	A group of fields that's used to partition the source data into partition groups.
`sourceName`	string	Required. Name of the source node.
`transformType`	BatchCalcJobTransformType (enumeration of type string)	Required. The type of transformation. Valid values are: • `ComputeRelative`—This transformation calculates values based on values of the same partition group. • `Expression`—This transformation calculates values based on existing values of fields in the same record. • `Slice`—This transformation removes fields from the source node.

BatchCalcJobTransformDroppedField

Represents a collection of fields relating to a dropped field in a data transformation of a data processing engine.

Fields

Field Name	Field Type	Description
`sourceFieldName`	string	Required. Name of the field that is dropped.

BatchCalcJobTransformAddedField

Represents a collection of fields relating to a formula in a data transformation of a data processing engine.

Fields

Field Name	Field Type	Description
`alias`	string	Required. Name that subsequent nodes within the data processing engine use to the transform node.
`dataType`	BatchCalcJobDataType (enumeration of type string)	Required. Data type of the formula. Valid values are: • `Text` • `Numeric` • `Date` • `DateTime` • `MultiValue`
`decimalPlaces`	integer	Number of digits to the right of a decimal point in the value. Required for the `Numeric` data type.
`expression`	string	Required. Formula defined by the user.
`length`	integer	Total length of the value including the decimal places. Required for data types: `Text` and `Numeric`.

BatchCalcJobOrderByField

Represents a collection of fields that are used to sort the partitioned data.

Fields

Field Name	Field Type	Description
`name`	string	Required. Name of the field that is used to sort data.
`orderType`	BatchCalcJobOrderType (enumeration of type string)	Order in which the data is sorted. Valid values are: • Ascending • Descending

BatchCalcJobUnion

Represents a collection of fields relating to the union of data from two nodes in a data processing engine.

Fields

Field Name	Field Type	Description
`description`	string	Description of the batch calculation job union.

Field Name	Field Type	Description
`isDisjointedSchema`	boolean	Indicates whether the union is of two disjointed datasets (`true`) or not (`false`). Set to `True` to allow joining of two datasets having no common fields.
`label`	string	Required. Label of the batch calculation job union.
`name`	string	Required. Name of the batch calculation job union.
`sources`	string[]	Names of the source nodes.

BatchCalcJobWritebackObject

Represents a collection of fields relating to the object in which the results of the data processing engine are written back.

Fields

Field Name	Field Type	Description
`canWrtbckToNonEditableFields`	boolean	Indicates whether the non-editable fields are included in field mapping when the action type is upsert. The default value is `false`. Available in API version 64.0 and later.
`description`	string	Descriptions of the batch calculation job writeback object.
`externalIdFieldName`	string	Unique external field ID for the target object name. Available in API version 60.0 and later.
`fields`	BatchCalcJobWriteback Mapping[]	Collection of the writeback fields.
`filterCondition`	string	The condition that filters the records from a writeback dataset for a user. Examples of a filter condition include a user ID, stage name, and a security policy that returns only the records that a user owns. Available in API version 57.0 and later.
`folderName`	string	The folder where the writeback dataset is saved. Available in API version 57.0 and later.
`isChangedRow`	boolean	Indicates whether a row in the write back object is changed. Set to `True` to write back the changed rows.
`isExistingDataset`	boolean	Indicates whether a CRM Application (CRMA) dataset or a Data Cloud Data Lake object is present (`true`) or will be created (`false`). Available in API version 62.0 and later.
`label`	string	Required. Name of the write back object.
`name`	string	Required. Name of the batch calculation job write back object.

Field Name	Field Type	Description
`operationType`	BatchCalcJobWritebackOpn (enumeration of type string)	Type of operation specified. Valid values are: • `Delete`—This value is available in API version 56.0 and later. • `Insert` • `Overwrite`—Available only when `storageType` is `DataLakeObject`. This value is available in API version 60.0 and later. • `Update` • `Upsert`
`sharingInheritanceObjectName`	string	The name of the source object from which the row-level sharing inheritance settings are applied. Available in API version 57.0 and later.
`sourceName`	string	Required. Name of the source node associated with the write back object.
`storageType`	BatchCalcJobWritebackType (enumeration of type string)	Specifies where you want to use the data stored in the source node. Available in API version 57.0 and later. Valid values are: • `Analytics` • `DataLakeObject` • `sObject` The default value is `sObject`.
`targetObjectName`	string	Required. Object that is inserted or upserted by the data processing engine.
`writebackSequence`	integer	Sequence in which the target object is updated by the data processing engine.
`writebackUser`	string	ID of the user whose permissions decide which objects and fields of the target object can be updated.

BatchCalcJobWritebackMapping

Represents a collection of fields relating to the mapping between results and the fields in the target object.

Fields

Field Name	Field Type	Description
`fieldType`	string	Target field type on the writeback object. Valid values are: • Primary Key • Qualifier Key Available in API version 64.0 and later.

Field Name	Field Type	Description
isAutogenerated	boolean	Indicates whether the target field value on the writeback object is autogenerated (`true`) or not (`false`). Available in API version 64.0 and later.
`parentName`	string	Name of the lookup object. Required only when the `relationshipName` field is defined.
`relationshipName`	string	Name of the lookup relationship.
`runtimeParameter`	boolean	Indicates whether the source field from runtime parameter is `true` or `false`. The default value is `false`. Available in API version 59.0 and later.
`sourceFieldName`	string	Required. Name of the field in the source node that is written back.
`targetFieldName`	string	Name of the sObject field to which the results are written back.

Declarative Metadata Sample Definition

The following is an example of a BatchCalcJobDefinition component.

```
<?xml version="1.0" encoding="UTF-8"?>
<BatchCalcJobDefinition xmlns="http://soap.sforce.com/2006/04/metadata">
    <aggregates>
        <description>Aggregate Description</description>
        <fields>
            <aggregateFunction>Count</aggregateFunction>
            <alias>NameCount</alias>
            <sourceFieldName>Name</sourceFieldName>
        </fields>
        <groupBy>ContactId</groupBy>
        <groupBy>Name</groupBy>
        <label>AggregateOpportunities</label>
        <name>AggregateOpportunities</name>
        <sourceName>Opportunity</sourceName>
    </aggregates>
    <forecasts>
        <description>ForecastNode Description</description>
        <label>ContactForecast</label>
        <name>ContactForecast</name>
        <sourceName>Contact</sourceName>
        <dateFieldName>CreatedDate</dateFieldName>
        <forecastPeriodType>YearMonth</forecastPeriodType>
        <shouldExcludeLastPeriod>false</shouldExcludeLastPeriod>
        <forecastPeriodCount>12</forecastPeriodCount>
        <periodStartDateName>CreatedDateYM</periodStartDateName>
        <forecastModelType>Auto</forecastModelType>
        <seasonality>None</seasonality>
        <accuracyPercent>None</accuracyPercent>
        <aggregationFields>
            <aggregateFunction>Count</aggregateFunction>
```

```
            <aggregationResultLabel>CountOfLastName</aggregationResultLabel>
            <fieldName>LastName</fieldName>
        </aggregationFields>
        <groupFields>
            <fieldName>LastModifiedDate</fieldName>
            <groupBy>Week</groupBy>
        </groupFields>
    </forecasts>
    <appends>
        <description>Append desc</description>
        <isDisjointedSchema>true</isDisjointedSchema>
        <label>AppendAllAccounts</label>
        <name>AppendAllAccounts</name>
        <sources>AccountsOfManufacturingIndustry</sources>
        <sources>ComputeRelativeManufacturingIndustry</sources>
    </appends>
    <datasources>
        <description>Desc Contact</description>
        <fields>
            <alias>Id</alias>
            <name>Id</name>
            <isPrimaryKey>false</isPrimaryKey>
            <dataType>Text</dataType>
        </fields>
        <fields>
            <alias>LastName</alias>
            <name>LastName</name>
            <isPrimaryKey>false</isPrimaryKey>
            <dataType>Text</dataType>
        </fields>
        <fields>
            <alias>CreatedDate</alias>
            <name>CreatedDate</name>
            <isPrimaryKey>false</isPrimaryKey>
            <dataType>Date</dataType>
        </fields>
        <fields>
            <alias>LastModifiedDate</alias>
            <name>LastModifiedDate</name>
            <isPrimaryKey>false</isPrimaryKey>
            <dataType>Date</dataType>
        </fields>
        <label>Contact</label>
        <name>Contact</name>
        <sourceName>Contact</sourceName>
        <type>StandardObject</type>
        <fileSource>ContentManagement</fileSource>
        <fileIdentifier>069xx0000004CAeAAM</fileIdentifier>
        <CSVDelimiter>COMMA</CSVDelimiter>
        <filePath>parentFolder/childFolder</filePath>
    </datasources>
    <datasources>
        <fields>
            <alias>Name</alias>
```

```
            <name>Name</name>
            <isPrimaryKey>false</isPrimaryKey>
            <dataType>Text</dataType>
        </fields>
        <fields>
            <alias>ContactId</alias>
            <name>ContactId</name>
            <isPrimaryKey>false</isPrimaryKey>
            <dataType>Text</dataType>
        </fields>
        <label>Opportunity</label>
        <name>Opportunity</name>
        <sourceName>Opportunity</sourceName>
        <type>StandardObject</type>
        <fileSource>ContentManagement</fileSource>
        <fileIdentifier>069xx0000004CAeAAM</fileIdentifier>
        <CSVDelimiter>COMMA</CSVDelimiter>
        <filePath>parentFolder/childFolder</filePath>
    </datasources>
    <description>Calculates and creates transaction journal records based on the orders
placed by the loyalty program members. The transaction journals are used to accrue points
 to the member.</description>
    <filters>
        <criteria>
            <operator>Equals</operator>
            <sequence>1</sequence>
            <sourceFieldName>LastName</sourceFieldName>
            <value>Salesforce</value>
        </criteria>
        <description>Filter Desc</description>
        <filterCondition>1</filterCondition>
        <isDynamicFilter>false</isDynamicFilter>
        <label>AccountsOfManufacturingIndustry</label>
        <name>AccountsOfManufacturingIndustry</name>
        <sourceName>AccountOpportunities</sourceName>
    </filters>
    <hierarchyPaths>
        <description>Hierarchy Path Node</description>
        <hierarchyFieldName>Hierarchy_Path</hierarchyFieldName>
        <isSelfFieldValueIncluded>true</isSelfFieldValueIncluded>
        <label>Get Hierarchy</label>
        <name>Get_Hierarchy</name>
        <parentFieldName>ContactId</parentFieldName>
        <selfFieldName>LastName</selfFieldName>
        <sourceName>AppendAllAccounts</sourceName>
    </hierarchyPaths>
    <isTemplate>false</isTemplate>
    <joins>
        <description>Left Outer Join</description>
        <fields>
            <alias>ContactId</alias>
            <sourceFieldName>Id</sourceFieldName>
            <sourceName>Contact</sourceName>
        </fields>
```

```
        <fields>
            <alias>LastName</alias>
            <sourceFieldName>LastName</sourceFieldName>
            <sourceName>Contact</sourceName>
        </fields>
        <fields>
            <alias>NameCount</alias>
            <sourceFieldName>NameCount</sourceFieldName>
            <sourceName>AggregateOpportunities</sourceName>
        </fields>
        <fields>
            <alias>OpportunityName</alias>
            <sourceFieldName>Name</sourceFieldName>
            <sourceName>AggregateOpportunities</sourceName>
        </fields>
        <joinKeys>
            <primarySourceFieldName>Id</primarySourceFieldName>
            <secondarySourceFieldName>ContactId</secondarySourceFieldName>
        </joinKeys>
        <label>AccountOpportunities</label>
        <name>AccountOpportunities</name>
        <primarySourceName>Contact</primarySourceName>
        <secondarySourceName>AggregateOpportunities</secondarySourceName>
        <type>LeftOuter</type>
    </joins>
    <label>Create Transaction Journals Based on Orders</label>
    <parameters>
        <dataType>Date</dataType>
        <defaultValue>2020-01-01</defaultValue>
        <description>Desc TextParameter</description>
        <isMultiValue>false</isMultiValue>
        <label>DateParameter</label>
        <name>DateParameter</name>
    </parameters>
    <parameters>
        <dataType>Filter</dataType>
        <defaultValue>{&quot;filterCondition&quot;: &quot;1 AND 2&quot;,
&quot;criteria&quot;: [{&quot;sourceFieldName&quot;:
&quot;NameCount&quot;,&quot;operator&quot;: &quot;GreaterThan&quot;,&quot;value&quot;:
&quot;20&quot;,&quot;sequence&quot;: &quot;1&quot;}, {&quot;sourceFieldName&quot;:
&quot;Name&quot;,&quot;operator&quot;: &quot;Equals&quot;,&quot;value&quot;:
&quot;Salesforce&quot;,&quot;sequence&quot;: &quot;2&quot;}]}</defaultValue>
        <isMultiValue>false</isMultiValue>
        <label>FilterParameter</label>
        <name>FilterParameter</name>
    </parameters>
    <parameters>
        <dataType>Numeric</dataType>
        <defaultValue>5000</defaultValue>
        <description>Desc TextParameter</description>
        <isMultiValue>false</isMultiValue>
        <label>NumericParameter</label>
        <name>NumericParameter</name>
    </parameters>
```

```
    <parameters>
        <dataType>Text</dataType>
        <defaultValue>@salesforce.com</defaultValue>
        <description>Desc TextParameter</description>
        <isMultiValue>false</isMultiValue>
        <label>TextParameter</label>
        <name>TextParameter</name>
    </parameters>
    <processType>Rebates</processType>
    <definitionRunMode>Batch</definitionRunMode>
    <status>Inactive</status>
    <transforms>
        <description>transforms Desc</description>
        <expressionFields>
            <alias>NewLastName</alias>
            <dataType>Text</dataType>
            <expression>TODAY()</expression>
            <length>80</length>
        </expressionFields>
        <label>ManufacturingIndustry</label>
        <name>ManufacturingIndustry</name>
        <sourceName>AccountsOfManufacturingIndustry</sourceName>
        <transformationType>Expression</transformationType>
    </transforms>
    <transforms>
        <droppedFields>
            <sourceFieldName>NewLastName</sourceFieldName>
        </droppedFields>
        <label>MediaIndustry</label>
        <name>MediaIndustry</name>
        <sourceName>ManufacturingIndustry</sourceName>
        <transformationType>Slice</transformationType>
    </transforms>
    <transforms>
        <description>compute relative transforms Desc</description>
        <expressionFields>
            <alias>NewLastName</alias>
            <dataType>Text</dataType>
            <expression>rank()</expression>
            <length>80</length>
        </expressionFields>
        <label>ComputeRelativeManufacturingIndustry</label>
        <name>ComputeRelativeManufacturingIndustry</name>
        <orderBy>
            <name>LastName</name>
            <orderType>Ascending</orderType>
        </orderBy>
        <partitionBy>LastName</partitionBy>
        <sourceName>MediaIndustry</sourceName>
        <transformationType>ComputeRelative</transformationType>
    </transforms>
    <customNodes>
        <name>RebatesCustomNode</name>
        <label>Rebates Custom Node</label>
```

```
        <description>customNodes Desc</description>
        <sources>Get_Hierarchy</sources>
        <extensionName>RebatesExpression</extensionName>
        <extensionNamespace>industries_mfg</extensionNamespace>
        <parameters>
            <name>inputColumn</name>
            <value>LastName</value>
        </parameters>
        <parameters>
            <name>isFilterCriteria</name>
            <value>true</value>
        </parameters>
        <parameters>
            <name>outputColumn</name>
            <value>GenName</value>
        </parameters>
    </customNodes>
    <writebacks>
        <fields>
            <sourceFieldName>GenName</sourceFieldName>
            <targetFieldName>LastName</targetFieldName>
        </fields>
        <isChangedRow>false</isChangedRow>
        <label>exportToContact</label>
        <name>exportToContact</name>
        <description>Export To Contact</description>
        <operationType>Insert</operationType>
        <sourceName>RebatesCustomNode</sourceName>
        <targetObjectName>Contact</targetObjectName>
        <writebackSequence>1</writebackSequence>
    </writebacks>
    <writebacks>
        <fields>
            <sourceFieldName>CreatedDateYM</sourceFieldName>
            <targetFieldName>CreatedDate</targetFieldName>
        </fields>
        <isChangedRow>false</isChangedRow>
        <isExistingDataset>false</isExistingDataset>
        <label>exportToContactFC</label>
        <name>exportToContactFC</name>
        <description>Export To Contact</description>
        <operationType>Insert</operationType>
        <sourceName>ContactForecast</sourceName>
        <targetObjectName>Contact</targetObjectName>
        <writebackSequence>2</writebackSequence>
    </writebacks>
</BatchCalcJobDefinition>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
   ~ Copyright 2020 Salesforce, Inc.
   ~ All Rights Reserved
   ~ Company Confidential
```

```
-->
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
   <types>
      <members>*</members>
      <name>BatchCalcJobDefinition</name>
   </types>
   <version>60.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

Data Processing Engine Invocable Actions

Run an active Data Processing Engine definition. For more information on custom invocable actions, see **REST API Developer Guide** and **Actions Developer Guide**.

Data Processing Engine Actions

Run an active Data Processing Engine definition. This action executes a Data Processing Engine definition asynchronously.

Data Processing Engine Actions

Run an active Data Processing Engine definition. This action executes a Data Processing Engine definition asynchronously.

A Data Processing Engine definition transforms data from your Salesforce org and writes back the results to your org. For more information about running Data Processing Engine definitions, see Run a Data Processing Engine Definition in Salesforce Help.

This object is available in API version 51.0 and later.

Supported REST HTTP Methods

URI

`/services/data/v`***`XX.X`***`/actions/custom/dataProcessingEngineAction`

Formats

JSON

HTTP Methods

GET, POST

Authentication

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

Inputs

Use the GET method to retrieve input variables of a Data Processing Engine definition. The input variables of each Data Processing Engine definition are unique. The Data Processing Engine action uses the input variables to execute the Data Processing Engine definition and generate a batch job ID.

Note: You can't use this action to start two simultaneous runs of a definition with the same input variables.

Outputs

Output	Details
`batchJobId`	**Type** string **Description** ID of the batch job generated after the request is successful. This ID is used to track the progress of the batch job in Monitor Workflow Services.

Usage

JSON Sample Request to execute PointsAccrual DPE Definition

```
{
   "PointsAccrual" : {
      "memberTier" : "Gold",
      "minimumPointBalanceRequired" : "50000",
      "pointType" : "non-qualifying"
   }
}
```

JSON Sample Response

```
{
   "actionName":"PointsAccrual",
      "errors":null,
      "isSuccess":true,
      "outputValues":{
        "batchJobId":"0lMxx0000A000001EAA"
      }
}
```

Example

GET

This example shows how to retrieve input variables of a Data Processing Engine action type.

```
curl --include --request GET \
--header "Authorization: Authorization: Bearer 00DR...xyz" \
--header "Content-Type: application/json" \
"https://instance.salesforce.com/services/data/v60.0/actions/custom/dataProcessingEngineAction/newinputvardefn"
```

POST

Here's a request to retrieve DPE definition

```
{
  "inputs": [
    {
      "start_date": "26-09-2023",
      "end_date": "12-12-2023",
      "randomkey": "069SM0000001SgbYAE"
```

```
    }
  ]
}
```

Here's a response for this action.

```
[
  {
    "actionName": "newinputvardefn",
    "errors": null,
    "invocationID": null,
    "isSuccess": true,
    "outputValues": {
      "batchJobId": "0mdSM0000006EJdYAM",
      "accepted": true
    },
    "version": 1
  }
]
```

Batch Management

Automate the processing of records in scheduled flows. You can process a high volume of standard and custom object records. Batch Management consists of three Tooling API objects, a standard object, a Metadata API, and an invocable action. You can use these resources to view, create, edit, and run Batch Management jobs.

EDITIONS

Available in: Lightning Experience

Available in: Batch Management is available with Enterprise, Unlimited, and Performance Editions with the Loyalty Management, Manufacturing Cloud, Rebate Management, or Accounting Subledger

Batch Management Tooling API Objects

You can use the Batch Management Tooling API object to view the settings of Batch Management jobs.

Batch Management Standard Objects

Batch Management contains a standard BatchProcessJobDefinitionView object, which you can use to view all the batch jobs available in your Salesforce org, including file-based jobs.

Batch Management Metadata API

Use a Metadata API to create, update, and activate Batch Management jobs.

Batch Management Invocable Actions

Run an active Batch Management job definition. For more information on custom invocable actions, see **REST API Developer Guide** and **Actions Developer Guide**.

Batch Management Tooling API Objects

You can use the Batch Management Tooling API object to view the settings of Batch Management jobs.

Tooling API objects let you interact with metadata for declarative development. For example, you can create your own version of Setup.

BatchProcessJobDefinition
Represents the details of a Batch Management job. This object is available in API version 51.0 and later.

EDITIONS

Available in: Lightning Experience

Available in: Batch Management is available with Enterprise, Unlimited, and Performance Editions with the Loyalty Management, Manufacturing Cloud, Rebate Management, or Accounting Subledger

BatchProcessJobDefinition

Represents the details of a Batch Management job. This object is available in API version 51.0 and later.

Note: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Data and processes in your org are impacted if you update or delete a BatchProcessJobDefinition record. Update or delete a Batch Management job using the Metadata API on page 157.

Supported SOAP API Calls

`describeSObjects()`, `query()`, `retrieve()`

Supported REST API Methods

`DELETE, GET, HEAD, PATCH, POST, Query`

Fields

Field	Details
`BatchJobDefinitionId`	**Type** reference **Properties** Filter, Group, Nillable, Sort **Description** The unique identifier of the associated batch job definition. This is a relationship field. **Relationship Name** BatchJobDefinition **Relationship Type** Lookup **Refers To** BatchJobDefinition

Field	Details
`BatchJobDefinitionName`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The name of the associated batch job definition.
`BatchSize`	**Type** int **Properties** Filter, Group, Sort **Description** Required. The number of records that each Batch Management job part can process. The maximum number of transaction journal records that a batch management job can process for flow or loyalty program process is 2000.
`Description`	**Type** textarea **Properties** Filter, Group, Nillable, Sort **Description** The description of the Batch Management job.
`DeveloperName`	**Type** string **Properties** Filter, Group, Sort **Description** The developer name of the Batch Management job.
`FlowDefinitionId`	**Type** reference **Properties** Filter, Group, Nillable, Sort **Description** The input variable of the associated flow that uniquely identifies each record that the Batch Management job processes.
`FullName`	**Type** string **Properties** Create, Group, Nillable

Field	Details
	Description The name of the Batch Management job. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.
`Language`	**Type** picklist **Properties** Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort **Description** The language in which the batch job is created. Possible values are: • `da`—Danish • `de`—German • `en_US`—English • `es`—Spanish • `es_MX`—Spanish (Mexico) • `fi`—Finnish • `fr`—French • `it`—Italian • `ja`—Japanese • `ko`—Korean • `nl_NL`—Dutch • `no`—Norwegian • `pt_BR`—Portuguese (Brazil) • `ru`—Russian • `sv`—Swedish • `th`—Thai • `zh_CN`—Chinese (Simplified) • `zh_TW`—Chinese (Traditional)
`ManageableState`	**Type** ManageableState enumerated list **Properties** Filter, Group, Nillable, Restricted picklist, Sort **Description** Indicates the manageable state of the specified component that is contained in a package: • `beta`

Field	Details
	• `deleted` • `deprecated` • `deprecatedEditable` • `installed` • `installedEditable` • `released` • `unmanaged`
`MasterLabel`	**Type** string **Properties** Filter, Group, Sort **Description** The label of the Batch Management job.
`Metadata`	**Type** complexvalue **Properties** Create, Nillable, Update **Description** The Batch Management job's metadata. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.
`NamespacePrefix`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The namespace prefix that is associated with this object. Each Developer Edition org that creates a managed package has a unique namespace prefix. Limit: 15 characters. You can refer to a component in a managed package by using the ***`namespacePrefix__componentName`*** notation. The namespace prefix can have one of the following values. • In Developer Edition orgs, `NamespacePrefix` is set to the namespace prefix of the org for all objects that support it, unless an object is in an installed managed package. In that case, the object has the namespace prefix of the installed managed package. This field's value is the namespace prefix of the Developer Edition org of the package developer.

Field	Details
	• In orgs that are not Developer Edition orgs, `NamespacePrefix` is set only for objects that are part of an installed managed package. All other objects have no namespace prefix.
`ProcessGroup`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The group or team for which the Batch Management job processes records.
`RecordIdVariable`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The unique identifier that identifies each record that must be processed by the Batch Management job.
`RetryCount`	**Type** int **Properties** Defaulted on create, Filter, Group, Sort **Description** Required. The number of times this Batch Management job must be rerun in case it fails.
`RetryInterval`	**Type** int **Properties** Defaulted on create, Filter, Group, Sort **Description** Required. The number of milliseconds after which the Batch Management job must be rerun in case it fails. A retry interval can be 1,000–10,000 milliseconds.
`Status`	**Type** picklist **Properties** Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort **Description** The status of the Batch Management job. Possible values are: • `Active`

Field	Details
	• `Inactive`
`Type`	**Type** picklist **Properties** Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort **Description** The type of process for which the Batch Management job processes records. Possible values are: • `DecisionTableRefresh` • `Flow` • `LoyaltyProgramProcess` The default value is `Flow`. This field is available in API version 55.0 and later.
`TypeInstance`	**Type** string **Properties** Filter, Group, Sort **Description** Required. The API name of the process that the Batch Management job must execute.

Batch Management Standard Objects

Batch Management contains a standard BatchProcessJobDefinitionView object, which you can use to view all the batch jobs available in your Salesforce org, including file-based jobs.

BatchProcessJobDefView
Represents the details of a Batch Job definition. The definition can also be file-based definitions that are available in your Salesforce org. This object is available in API version 51.0 and later.

EDITIONS

Available in: Lightning Experience

Available in: Batch Management is available with Enterprise, Unlimited, and Performance Editions with the Loyalty Management, Manufacturing Cloud, Rebate Management, or Accounting Subledger

BatchProcessJobDefView

Represents the details of a Batch Job definition. The definition can also be file-based definitions that are available in your Salesforce org. This object is available in API version 51.0 and later.

Supported Calls

`describeSObjects(),query()`

Fields

Field	Details
`DurableId`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The unique identifier for the field. Always retrieve this value before using it, as the value isn't guaranteed to stay the same from one release to the next. Simplify queries by using this field instead of making multiple queries.
`IsActive`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** Indicates whether the definition is active.
`Label`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The label of the Batch Job definition.
`Name`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The name of the Batch Job definition.
`NamespacePrefix`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The namespace prefix associated with this object. Each Developer Edition organization that creates a managed package has a unique namespace prefix. Limit: 15 characters. You can refer to a component in a managed package by using the namespacePrefix__componentName notation. The namespace prefix can have one of the following values:

Field	Details
	• In Developer Edition organizations, the namespace prefix is set to the namespace prefix of the organization for all objects that support it. There is an exception if an object is in an installed managed package. In that case, the object has the namespace prefix of the installed managed package. This field's value is the namespace prefix of the Developer Edition organization of the package developer. • In organizations that are not Developer Edition organizations, NamespacePrefix is only set for objects that are part of an installed managed package. There is no namespace prefix for all other objects.
`ProcessDefinition`	**Type** textarea **Properties** Nillable **Description** The name of the process group for the batch process job definition.
`ProcessGroup`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The team or group for which the definition processes records.
`SourceObjectName`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The API name of the object whose records are processed.
`Type`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The type of process for which the Batch Management job processes records. Possible values are: • `Flow` • `LoyaltyProgramProcess` This field is available in API version 55.0 and later.

Field	Details
`TypeInstance`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The API name of the process that's processed by the Batch Job definition.

Batch Management Metadata API

Use a Metadata API to create, update, and activate Batch Management jobs.

BatchProcessJobDefinition
Represents the details of a Batch Management job definition. This type extends the Metadata metadata type and inherits its `fullName` field.

EDITIONS

Available in: Lightning Experience

Available in: Batch Management is available with Enterprise, Unlimited, and Performance Editions with the Loyalty Management, Manufacturing Cloud, Rebate Management, or Accounting Subledger

BatchProcessJobDefinition

Represents the details of a Batch Management job definition. This type extends the Metadata metadata type and inherits its `fullName` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

BatchProcessJobDefinition components have the suffix `.batchProcessJobDefinition` and are stored in the `batchProcessJobDefinitions` folder.

Version

BatchProcessJobDefinition components are available in API version 51.0 and later.

Special Access Rules

To use this metadata type, your Salesforce org must have the Loyalty Management or the Rebate Management license. The Loyalty Program Process type is only available in orgs that have Loyalty Management enabled.

Fields

Field Name	Field Type	Description
`batchSize`	integer	Required. Number of records that each Batch Management job can process. Flow type Batch Management jobs can process up to 2000

Field Name	Field Type	Description
		records and Loyalty Program Process type Batch Management jobs can process up to 250 records.
`dataSource`	BatchDataSource on page 159[]	Required. Source of information whose records must be processed by the Batch Management job.
`description`	string	Description of the Batch Management job, up to 255 characters.
`executionProcessApiName`	string	API name of process that must be executed by the Batch Management job. This field is available in API version 55.0 and later. • If the batch job's type is Flow, enter the API name of an active flow that the batch job must execute. • If the batch job's type is Loyalty Program Process, enter: – Transaction_Journals if you want the batch job to process Transaction Journal records by applying the applicable active loyalty program process of the type TransactionJournal. – API name of an active loyalty process of the type TierProcessing if you want the batch job to run the loyalty program process to assess the tier of eligible members. The API name consists of the name of the process, the process type, and the name of the loyalty program separated by two consecutive underscores. For example, the process API name is `Update Member Tier__TierProcessing__Inner Circle` if the process name is Update Member Tier, the process type is TierProcessing, and the loyalty program name is Inner Circle. You can use database-based APEX classes that let you use flex queues in the Batch Management job, allowing to place more than 5 jobs in a queue. This functionality is applicable to all Industry Clouds that use managed packages. See Apex Flex Queue.
`flowApiName`	string	API name of an active flow process that must be executed by the Batch Management job. Note: You can either specify the flow API name in the `executionProcessApiName` field or in the `flowApiName` field.
`flowInputVariable`	string	Input variable of associated flow that is used by the batch job to uniquely identify records.
`masterLabel`	string	Required. Name of the Batch Management job, up to 80 characters.
`processGroup`	string	Required. Name of the group for which the Batch Management job processes records.
`retryCount`	integer	Required. Number of times this Batch Management job must be rerun in case it fails. The maximum retry count is 3. Valid values are 1–3.

Field Name	Field Type	Description
`retryInterval`	integer	Required. Number of milliseconds after which the Batch Management job must be rerun in case it fails. Valid values are 1,000–10,000.
`status`	string	Indicates the status of the Batch Management job. Valid values are `Active` and `Inactive`.
`type`	string (enumeration of type string)	The type of process that the Batch Management job must execute. This field is available in API version 55.0 and later. Valid values are: • `Flow` • `Loyalty Program Process`

BatchDataSource

Represents the source of information whose records must be processed by the Batch Management job.

Fields

Field Name	Field Type	Description
`condition`	string	Required. Criteria defined to filter the records.
`criteria`	string	Type of filter criteria that's used to filter records for processing.
`dataSourceType`	string	Type of data source that's used to create the batch job definition. Valid values are: • SingleSobject • MultiSobject Available in API version 64.0 and later.
`filters`	BatchDataSrcFilterCriteria on page 160[]	Filter criterion that decides which records must be processed by the Batch Management job.
orderFields	BatchDataSourceOrderField on page 161	Fields that are used to order the records before the records are added to a batch in a job.
`sourceObject`	string	Required. API name of an object whose records must be processed by the batch job. If the batch job type is Loyalty Program Process, the source object must be: • TransactionJournal if the batch job is used to process transaction journals by applying the applicable loyalty program process. • An object that stores the details of loyalty program members whose tier must be assessed by the loyalty program process specified in the executionProcessApiName field.

Field Name	Field Type	Description
`sourceObjectField`	string	API name of the source object field that uniquely identifies records for which the batch job is executed. This field is available in API version 57.0 and later. This field is only applicable when the batch job's type is Loyalty Program Process and a TierProcess type active loyalty program process is specified in the `executionProcessApiName` field. Specify the API name of a field that is a lookup to the LoyaltyProgramMember object and uniquely identifies the members whose tier must be assessed.

BatchDataSrcFilterCriteria

Represents the filter conditions that decide which records must be processed by the Batch Management job.

Fields

Field Name	Field Type	Description
domainObjectName	string	Name of the object the field is associated with. Available in API version 64.0 and later.
`dynamicValueType`	string	Data type of the input variable used as a filter.
`fieldName`	string	Required. Name of the field that must be used to filter records.
`fieldPath`	string	Stores the path to a field in the object. Available in API version 64.0 and later.
`fieldValue`	string	Required. Value of the field that must be filtered. Specify the field if `isDynamicValue` is set to `False`.
`isDynamicValue`	boolean	Required. Indicates whether the filter criteria is dynamic.
`operator`	string (enumeration of type string)	Required. Operator that is specified in the filter criteria. Valid values are: • `equals` • `excludes` • `greaterThan` • `greaterThanOrEqualTo` • `in` • `includes` • `lessThan` • `LessThanOrEqualTo` • `GreaterOrEqual` • `like` • `notEquals` • `notIn`
`sequenceNo`	integer	Required. Sequence number used to refer the criteria in a filter.

BatchDataSourceOrderField

Represents the fields that are used to group data.

Fields

Field Name	Field Type	Description
domainObjectName	string	Required. Name of the object the field is associated with. Available in API version 64.0 and later.
`fieldName`	string	Required. Name of the field that must be used to filter records. Available in API version 64.0 and later.
`fieldPath`	string	Required. Stores the path to a field in the object. Available in API version 64.0 and later.

Declarative Metadata Sample Definition

The following is an example of a BatchProcessJobDefinition component.

```
<?xml version="1.0" encoding="UTF-8"?>
<BatchProcessJobDefinition xmlns="http://soap.sforce.com/2006/04/metadata">
   <batchSize>10</batchSize>
   <dataSource>
      <condition>1</condition>
      <criteria>all</criteria>
      <filters>
         <dynamicValue>false</dynamicValue>
         <dynamicValueType>string</dynamicValueType>
         <fieldName>Name</fieldName>
         <fieldValue>abcd</fieldValue>
         <operator>equals</operator>
         <sequenceNo>1</sequenceNo>
      </filters>
      <sourceObject>Account</sourceObject>
   </dataSource>
   <flowApiName>Flow1</flowApiName>
   <flowInputVariable>recordId</flowInputVariable>
   <masterLabel>BatchJob1</masterLabel>
   <processGroup>Loyalty</processGroup>
   <retryCount>2</retryCount>
   <retryInterval>1000</retryInterval>
   <status>Inactive</status>
   <description>test</description>
   <type>Flow</type>
   <executionProcessApiName>testFlow</executionProcessApiName>
</BatchProcessJobDefinition>
```

The following is an example of a Flow object used in Metadata API.

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!--
   ~ Copyright 2020 Salesforce, Inc.
   ~ All Rights Reserved
   ~ Company Confidential
-->
<Flow xmlns="http://soap.sforce.com/2006/04/metadata">
   <apiVersion>51.0</apiVersion>
   <interviewLabel>Flow1 {!$Flow.CurrentDateTime}</interviewLabel>
   <label>Flow1</label>
   <processMetadataValues>
      <name>BuilderType</name>
      <value>
         <stringValue>LightningFlowBuilder</stringValue>
      </value>
   </processMetadataValues>
   <processMetadataValues>
      <name>OriginBuilderType</name>
      <value>
         <stringValue>LightningFlowBuilder</stringValue>
      </value>
   </processMetadataValues>
   <processType>AutoLaunchedFlow</processType>
   <recordLookups>
      <name>getAcc</name>
      <label>getAcc</label>
      <locationX>614</locationX>
      <locationY>465</locationY>
      <assignNullValuesIfNoRecordsFound>false</assignNullValuesIfNoRecordsFound>
      <filterLogic>and</filterLogic>
      <filters>
         <field>Id</field>
         <operator>EqualTo</operator>
         <value>
            <elementReference>recordId</elementReference>
         </value>
      </filters>
      <getFirstRecordOnly>true</getFirstRecordOnly>
      <object>Account</object>
      <storeOutputAutomatically>true</storeOutputAutomatically>
   </recordLookups>
   <start>
      <locationX>73</locationX>
      <locationY>213</locationY>
      <connector>
         <targetReference>getAcc</targetReference>
      </connector>
   </start>
   <status>Draft</status>
   <variables>
      <name>recordId</name>
      <dataType>String</dataType>
      <isCollection>false</isCollection>
      <isInput>true</isInput>
      <isOutput>false</isOutput>
```

```
    </variables>
</Flow>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>BatchProcessJobDefinition</name>
    </types>
    <types>
        <members>Flow1</members>
        <name>Flow</name>
    </types>
    <version>51.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

Batch Management Invocable Actions

Run an active Batch Management job definition. For more information on custom invocable actions, see **REST API Developer Guide** and **Actions Developer Guide**.

Batch Job Actions

Run an active Batch Management job definition. This action executes a defined Batch Management job asynchronously.

Submit Failed Records Batch Job

Run to resubmit an existing batch job with failed records for processing. This action executes the batch job asynchronously.

Batch Job Actions

Run an active Batch Management job definition. This action executes a defined Batch Management job asynchronously.

A Batch Management job processes a flow in manageable parts. For more information about running an active Batch Management jobs, see Schedule a Batch Job in Salesforce Help.

This object is available in API version 51.0 and later.

Supported REST HTTP Methods

URI

`/services/data/v`***`XX.X`***`/actions/custom/batchJobAction`

Formats

JSON

HTTP Methods
GET, POST

Authentication
`Authorization: Bearer token`

Inputs

The batch job action uses the batch job definition ID and input variables to execute the job and generate a batch job ID. The input values vary according to the input variables in that flow.

Outputs

Output	Details
`batchJobId`	**Type** string **Description** ID of the batch job generated after the request is successful. This batch job ID is used to track the progress of the batch job in Monitor Workflow Services.

Usage

A request body is always required. The input values vary according to the input variables in that flow. If your batch job doesn't require any input variables, you still must send an empty JSON body.

```
{
  "inputs": [{}]
}
```

JSON Sample Request

```
{
   "noOfEmployees" : 900,
   "accountIndustry" : "Technology"
}
```

JSON Sample Response

```
{
   "batchJobId": "0lMxx0000A000001EAA"
   "accepted": "true"
}
```

Submit Failed Records Batch Job

Run to resubmit an existing batch job with failed records for processing. This action executes the batch job asynchronously.

This object is available in API version 52.0 and later.

Supported REST HTTP Methods

URI

`/services/data/v***XX.X***/actions/standard/submitFailedRecordsBatchJob`

Formats

JSON

HTTP Methods

GET, POST

Authentication

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

Inputs

Input	Details
`failedRecordIds`	**Type** array **Description** The IDs of failed records in a batch job.
`parentBatchJobId`	**Type** string **Description** Required. The ID of a batch job with failed records.

Outputs

Output	Details
`batchJobId`	**Type** string **Description** The ID of a batch job generated after the request is successful. This batch job ID is used to track the progress of the batch job in Monitor Workflow Services in the org.
`status`	**Type** string **Description** Indicates whether a batch job succeeded or failed.

Usage

JSON Sample Request

```
{
   "inputs": [{
      "parentBatchJobId": "0mdRM0000004DXrYAM",
      "failedRecordIds": [
         "001RM000005AG0bYAG", "001RM000005AERZYA4", "001RM000005AG0WYAW"
      ]
   }]
}
```

JSON Sample Response

```
[ {
   "actionName" : "submitFailedRecordsBatchJob",
   "errors" : null,
   "isSuccess" : true,
   "outputValues" : {
      "batchJobId" : "0mdRM0000004DZ9YAM"
   }
} ]
```

Monitor Workflow Services

The Montior Workflow Services standard objects can be used to track the run of Data Processing Engine definitons and Batch Management jobs. During a run, you can view details about each part that the run is broken down into. After the run is complete, you can view its status and the records which weren't processed during the run.

The objects of Monitor Workflow Services aren't available in Object Manager of your Salesforce org.

BatchJob
Represents an instance of a batch job that is either running and has been run. This object is available in API version 51.0 and later.

BatchJobPart
Represents one part of a batch job. This object is available in API version 51.0 and later.

BatchJobPartFailedRecord
Represents records that a batch job part couldn't successfully process. This object is available in API version 51.0 and later.

EDITIONS

Available in: Lightning Experience

Available in: Monitor Workflow Services is available with Enterprise, Unlimited, and Performance Editions where Data Processing Engine or Batch Management is avaiable

BatchJob

Represents an instance of a batch job that is either running and has been run. This object is available in API version 51.0 and later.

Supported Calls

`delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`

Fields

Field	Details
`BatchJobDefinitionId`	**Type** reference **Properties** Filter, Group, Sort **Description** The unique identifier of the associated batch job definition. This is a relationship field. **Relationship Name** BatchJobDefinition **Relationship Type** Lookup **Refers To** BatchJobDefinition
`BatchJobDefinitionName`	**Type** string **Properties** Filter, Group, Sort **Description** The developer name of the associated batch job definition.
`EndTime`	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** The timestamp when the batch job run was completed.
`ErrorDescription`	**Type** string **Properties** Filter, Nillable, Sort **Description** The error message in case the batch job run failed.
`ExternalReference`	**Type** string **Properties** Filter, Group, Nillable, Sort

Field	Details
	Description The unique identifier of the process that's running or has run the batch job.
`IsDebugOn`	**Type** boolean **Properties** Create, Defaulted on create, Filter, Group, Sort, Update **Description** The default value is `false`.
`IsDebugRecipeDeleted`	**Type** boolean **Properties** Create, Defaulted on create, Filter, Group, Sort, Update **Description** The default value is `false`.
`LastReferencedDate`	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** The timestamp for when the current user last viewed the batch job.
`LastViewedDate`	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** The timestamp for when the current user last viewed this item.
`Name`	**Type** string **Properties** Filter, Group, idLookup, Sort **Description** The name of the batch job.
`OwnerId`	**Type** reference

Field	Details
	Properties Filter, Group, Sort **Description** Unique identifier of the user who initiated the batch job run. This is a polymorphic relationship field. **Relationship Name** Owner **Relationship Type** Lookup **Refers To** Group, User
`ProcessGroup`	**Type** string **Properties** Filter, Group, Sort **Description** The group or team for which the batch job is run.
`RetryCount`	**Type** int **Properties** Filter, Group, Nillable, Sort **Description** The number of times the batch job run is automatically rerun in case it fails.
`RuntimeParameter`	**Type** textarea **Properties** Nillable **Description** The values of the input variables that are used as filter criteria in a Batch Management job.
`StartTime`	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** The timestamp when the batch job run was started.

Field	Details
`Status`	**Type** picklist **Properties** Filter, Group, Restricted picklist, Sort **Description** The status of the batch job run. Possible values are: • `Canceled` • `Completed` • `CompletedWithFailures` • `Failed` • `InProgress` • `Submitted`
`Type`	**Type** picklist **Properties** Defaulted on create, Filter, Group, Restricted picklist, Sort **Description** The type of batch job that is run. Possible values are: • `Calc`—Data Processing Engine • `DecisionTableRefresh` • `Flow`

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

BatchJobFeed
Feed tracking is available for the object.

BatchJobHistory
History is available for tracked fields of the object.

BatchJobPart

Represents one part of a batch job. This object is available in API version 51.0 and later.

When a batch job is run, it is divided in to multiple parts. Each part is used to process a specific number of records.

Supported Calls

`describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`

Fields

Field	Details
`BatchJobId`	**Type** reference **Properties** Filter, Group, Sort **Description** The unique identifier of the associated batch job. This is a relationship field. **Relationship Name** BatchJob **Relationship Type** Lookup **Refers To** BatchJob
`EndTime`	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** The timestamp when the batch job part was processed.
`ErrorDescription`	**Type** string **Properties** Filter, Nillable, Sort **Description** The error message in case the batch job part failed.
`FailedRecordCount`	**Type** int **Properties** Filter, Group, Nillable, Sort **Description** The number of records that the batch job part couldn't process.

Field	Details
`FailedRecFile`	**Type** FILE **Properties** Filter, Group, Sort **Description** The file containing failed records downloaded as CSV, JSON, or TXT formats during a Data Processing Engine definition's run by using CRM Analytics (CRMA) or Data Cloud runtime. Available in API version 63.0 and later.
`InputRecordCount`	**Type** int **Properties** Filter, Group, Nillable, Sort **Description** The number of records that the batch job part must process.
`Name`	**Type** string **Properties** Filter, Group, idLookup, Sort **Description** The name of the batch job part.
`OutputRecordCount`	**Type** int **Properties** Filter, Group, Nillable, Sort **Description** The number of records the batch job part has processed.
`ParentBatchJobPartId`	**Type** reference **Properties** Filter, Group, Nillable, Sort **Description** The unique identifier of the part batch job part associated with the batch job part. This is a relationship field. **Relationship Name** ParentBatchJobPart **Relationship Type** Lookup

Field	Details
	Refers To BatchJobPart
`RetryCount`	**Type** int **Properties** Filter, Group, Nillable, Sort **Description** The number of times the batch job part is automatically rerun in case it fails.
`StartTime`	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** The timestamp when the batch job part's run was started.
`Status`	**Type** picklist **Properties** Defaulted on create, Filter, Group, Restricted picklist, Sort **Description** The status of the batch job part. Possible values are: • `Canceled`—Canceled • `Completed` • `Failed` • `InProgress`—In Progress
`Type`	**Type** picklist **Properties** Filter, Group, Nillable, Restricted picklist, Sort **Description** The type of node in case the associated batch job is of the type Calc (Data Processing Engine). Possible values are: • `Aggregate` • `Append` • `Compute` • `Custom`

Field	Details
	• `Datasync` • `Filter` • `Forecast` • `Hierarchy` • `Join` • `Slice` • `Source` • `Transform` • `Writeback`

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

BatchJobPartFeed
Feed tracking is available for the object.

BatchJobPartHistory
History is available for tracked fields of the object.

BatchJobPartFailedRecord

Represents records that a batch job part couldn't successfully process. This object is available in API version 51.0 and later.

Supported Calls

`describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`

Fields

Field	Details
`BatchJobId`	**Type** reference **Properties** Filter, Group, Nillable, Sort **Description** The unique identifier of the associated batch job. This is a relationship field. **Relationship Name** BatchJob

Field	Details
	Relationship Type Lookup **Refers To** BatchJob
`BatchJobPartId`	**Type** reference **Properties** Filter, Group, Sort **Description** The unique identifier of the associated batch job part. This is a relationship field. **Relationship Name** BatchJobPart **Relationship Type** Lookup **Refers To** BatchJobPart
`ErrorDescription`	**Type** string **Properties** Filter, Nillable, Sort **Description** The error message that indicates why the batch job part couldn't process the records.
`Name`	**Type** string **Properties** Filter, Group, idLookup, Sort **Description** The name of the batch job part failed record.
`Record`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The unique identifier of the batch record that processed the failed records.

Field	Details
`RecordName`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The name of the record that's associated with the batch job part failed record.
`ResubmittedBatchJobId`	**Type** reference **Properties** Filter, Group, Nillable, Sort **Description** The batch job used to submit failed records. This field is available in API version 52.0 and later. This is a relationship field. **Relationship Name** ResubmittedBatchJob **Relationship Type** Lookup **Refers To** BatchJob
`Status`	**Type** picklist **Properties** Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort **Description** Specifies the status of the failed records. This field is available in API version 52.0 and later. Possible values are: • `Failed` • `Resubmitted` The default value is 'Failed'.

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

BatchJobPartFailedRecordFeed
Feed tracking is available for the object.

BatchJobPartFailedRecordHistory

History is available for tracked fields of the object.