Analytics External Data API Developer Guide

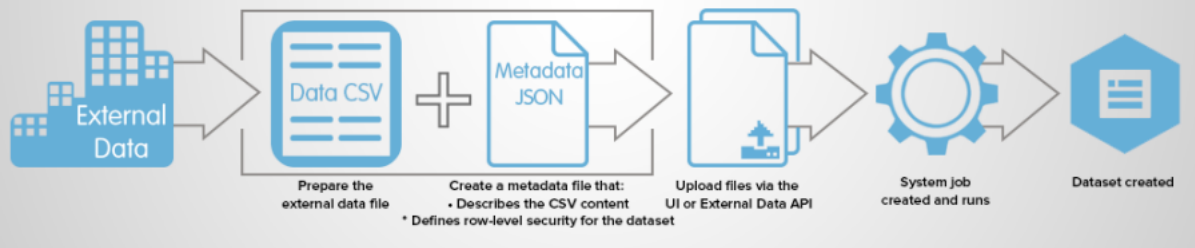
Salesforce, Summer '25

Last updated: June 20, 2025

CONTENTS

EXTERNAL DATA

You can integrate external data into CRM Analytics to make the data available for queries from lenses and designer.



The External Data API enables you to upload external data files to CRM Analytics. The External Data API can upload .csv files, and you can optionally specify the structure of your data by defining metadata in JSON format.

The External Data API is available in API version 31 and later.

The high-level steps for uploading external data by using the API are:

1. Prepare your data in CSV format, and then create a metadata file to specify the structure of the data.
2. Connect programmatically to your Salesforce organization.
3. Configure the upload by inserting a row into the InsightsExternalData object, and then set input values such as the name of the dataset, the format of the data, and the operation to perform on the data.
4. Split your data into 10-MB chunks, and then upload the chunks to InsightsExternalDataPart objects.
5. Start the upload by updating the `Action` field in the InsightsExternalData object.
6. Monitor the InsightsExternalData object for status updates, and then verify that the file upload was successful.

External Data Metadata Overview

To upload external data into a CRM Analytics dataset, make sure that you have prepared your data and metadata files.

Load External Data into a Dataset

External Data API Limits

When working with the External Data API, consider the following limits, in addition to the general CRM Analytics limits.

Value	Limit
Maximum file size per external data uploads	40 GB
Maximum file size for all external data uploads in a rolling 24-hour period	50 GB
Maximum number of external data jobs per dataset that can be run in a rolling 24-hour period	50

Value	Limit
Maximum number of characters in a field	32,000
Maximum number of fields in a record	5,000 (including up to 1,000 date fields)
Maximum number of characters for all fields in a record	400,000

For more information on limits for all supported editions, see External Data Limits in CRM Analytics Limits.

Prepare Data Files

To upload external data from .csv files into a dataset, first prepare your data files.

External data can be loaded into a dataset by preparing two files.

- A data file, which contains the external data, in comma-separated value (CSV) format
- An optional metadata .json file, which describes the structure of the data file

Note: Providing a metadata file is recommended.

The data and metadata files are used to populate a dataset with the external data. For detailed information about formatting CSV data and JSON metadata, see the External Data Metadata Overview.

Connect to Salesforce

After preparing your data files, the next step in loading external data into CRM Analytics is to connect to your Salesforce organization by using standard Salesforce APIs.

Note: The following examples use SOAP API, but you can use any of the Salesforce APIs, such as REST API or Apex. The examples assume that you're using the Web Services Connector.

To load external data into CRM Analytics, first connect to your Salesforce organization. Use the PartnerConnection object to log in to your organization, as shown in the following example. You need to supply a username, password, and endpoint.

```
ConnectorConfig config = new ConnectorConfig();
config.setUsername(username);
config.setPassword(password);
config.setAuthEndpoint(endpoint);

PartnerConnection partnerConnection = new PartnerConnection(config);
```

For more information about the Web Services Connector (WSC), see Introduction to the Web Services Connector. For more information about user authentication, see Security and the API in the *SOAP API Developer Guide*.

Configure the Upload

Configure the external data upload by inserting a row into the InsightsExternalData object and setting configuration values.

After establishing a connection with Salesforce, insert a row into the InsightsExternalData object to configure and control the upload. The InsightsExternalData object provides a "header" that contains information about the upload, such as the name of the dataset, the

format of the data, and the operation to perform on the data. You can also provide the metadata file. The following example inserts a row into the InsightsExternalData object and sets configuration values.

```
SObject sobj = new SObject();
sobj.setType("InsightsExternalData");
sobj.setField("Format","Csv");
sobj.setField("EdgemartAlias", DatasetName);
sobj.setField("MetadataJson",metadataJson);
sobj.setField("Operation","Overwrite");
sobj.setField("Action","None");

SaveResult[] results = partnerConnection.create(new SObject[] { sobj });

for(SaveResult sv:results)
    if(sv.isSuccess())
        parentID = sv.getId();
```

Note: The WSC converts the metadata .json file to a Base64-encoded string, but if you're using REST API, you need to make this conversion yourself.

For detailed information about the InsightsExternalData object, see The InsightsExternalData Object.

Add the Data

When uploading external data files, you can use the InsightsExternalDataPart object to load the data in smaller chunks.

After inserting a row into the InsightsExternalData (header) object, split your data into 10-MB chunks and upload the chunks to InsightsExternalDataPart objects. You associate the part objects with the header object by setting the `InsightsExternalDataId` field on the part objects to the ID of the header object. The part objects contain the bytes of data and must be assigned part numbers in a contiguous sequence, starting with 1.

Ensure that the chunks of data are smaller than 10 MB. If the data is compressed, it must be compressed first and then split into 10-MB chunks. Only the gzip format is supported.

The following example splits a file into 10-MB chunks, and then uploads the chunks to InsightsExternalDataPart objects.

```
List<File> fileParts = chunkBinary(dataFile); //Split the file

for(int i = 0;i<fileParts.size();i++)
{
    SObject sobj = new SObject();
    sobj.setType("InsightsExternalDataPart");
    sobj.setField("DataFile", FileUtils.readFileToByteArray(fileParts.get(i)));
    sobj.setField("InsightsExternalDataId", parentID);
    obj.setField("PartNumber",i+1); //Part numbers should start at 1
    SaveResult[] results = partnerConnection.create(new SObject[] { sobj });
    for(SaveResult sv:results)
        if(sv.isSuccess())
            rowId = sv.getId();
}
```

For detailed information about the InsightsExternalDataPart object, see InsightsExternalDataPart.

Manage the Upload

After you've created a header and uploaded the data parts by using the `InsightsExternalData` and InsightsExternalDataPart objects, update the `Action` field on the header object to `Process` to start processing the data.

The following example sets the `Action` field and updates the row in the InsightsExternalData object.

```
SObject sobj = new SObject();
sobj.setType("InsightsExternalData");
sobj.setField("Action","Process");
sobj.setId(parentID); // This is the rowID from the previous example.
SaveResult[] results = partnerConnection.update(new SObject[] { sobj });

for(SaveResult sv:results)
    if(sv.isSuccess())
        rowId = sv.getId();
```

When the `Action` field is set to `Process`, a dataflow job is created and marked active. You can monitor the `Status` field of the header object to determine when the file upload is completed. After the `Action` field is updated to request processing, no user edits are allowed on the objects.

Update the Data

After you upload data into a dataset, update it by adding, upserting, or removing data.

To update data, set the `Action` field to `Process` and the `Operation` field to `Append`, `Upsert`, or `Delete`.

This example sets the `Action` and `Operation` fields to append data to the InsightsExternalData object. To enable faster uploads, use the `Mode` field with a value of `Incremental`.

```
SObject sobj = new SObject();
sobj.setType("InsightsExternalData");
sobj.setField("InsightsExternalDataId", parentID);
sobj.setField("Action","Process");
sobj.setField("Operation","Append");
sobj.setField("Mode","Incremental");

SaveResult[] results = partnerConnection.update(new SObject[] { sobj });

for(SaveResult sv:results)
  if(sv.isSuccess())
    parentID = sv.getId();
```

If `Mode` isn't specified or is set to `None`, the append is a bulk upload that processes data slower.

External Data Metadata Overview

To upload external data into a CRM Analytics dataset, make sure that you have prepared your data and metadata files.

You can load external data into a dataset by preparing two files.

- A data file, which contains the external data, in comma-separated value (CSV) format
- An optional metadata file, which describes the structure of the data file in JSON format

Note: Providing a metadata file is recommended. Otherwise, every field is treated as text.

The data and metadata files are used to populate a dataset with the external data.

CSV Example

The following CSV example contains data that conforms to the .json metadata file that's described next.

```
Name,Amount,CloseDate
opportunityA,100.99,6/30/2014
opportunityB,99.01,1/31/2012
```

The first row in the CSV file lists the field names for your dataset. Each subsequent row corresponds to a record of data. A record consists of a series of fields delimited by commas. For information on creating valid field names, see External Data Metadata Format Reference.

JSON Example

The following JSON example represents a SalesData object with three fields: Name, Amount, and CloseDate. The example corresponds to the preceding CSV example.

```
{
    "fileFormat": {
        "charsetName": "UTF-8",
        "fieldsDelimitedBy": ",",
        "fieldsEnclosedBy": "\"",
        "fieldsEscapedBy":""
        "linesTerminatedBy":"\r\n"
        "numberOfLinesToIgnore": 1
    },
    "objects": [
        {
            "connector": "AcmeCSVConnector",
            "description": "",
            "fullyQualifiedName": "SalesData",
            "label": "Sales Data",
            "name": "SalesData",
            "fields": [
                {
                    "description": "",
                    "fullyQualifiedName": "SalesData.Name",
                    "label": "Account Name",
                    "name": "Name",
                    "isSystemField": false,
                    "isUniqueId": false,
                    "isMultiValue": false,
                    "type": "Text"
                },
                {
                    "description": "",
                    "fullyQualifiedName": "SalesData.Amount",
                    "label": "Opportunity Amount",
                    "name": "Amount",
                    "isSystemField": false,
                    "defaultValue": "0",
                    "isUniqueId": false,
```

```
                    "type": "Numeric",
                    "precision": 10,
                    "scale": 2,
                },
                {
                    "description": "",
                    "fullyQualifiedName": "SalesData.CloseDate",
                    "label": "Opportunity Close Date",
                    "name": "CloseDate",
                    "isSystemField": false,
                    "isUniqueId": false,
                    "type": "Date",
                    "format": "MM/dd/yyyy",
                    "fiscalMonthOffset": 0
                }
            ]
        }
    ]
}
```

CSV Format

The External Data API uses a strict format for field values to optimize processing for large sets of data. Note the following when generating .csv files.

- If a field value contains a control character or a new line the field value must be contained within double quotes (or your `fieldsEscapedBy` value). The default control characters (`fieldsDelimitedBy`, `fieldsEnclosedBy`, `fieldsEscapedBy`, or `linesTerminatedBy`) are comma and double quote. For example, `"Director of Operations, Western Region"`.
- If a field value contains a double quote, escape the double quote by preceding it with another double quote (or your `fieldsEscapedBy` value): for example, `"This is the ""gold"" standard"`.
- Field values aren't trimmed. A space before or after a delimiting comma is included in the field value. A space before or after a double quote generates an error for the row. For example, `John,Smith` is valid. `John, Smith` is valid, but the second value is `" Smith"`. `"John", "Smith"` isn't valid.
- The maximum numeric value is 36,028,797,018,963,967 and the minimum is -36,028,797,018,963,968.
- Dates must conform to specific formats, and they must match the formats exactly. For more information, see External Data Metadata Format Reference.
- At least one column in the CSV file must contain dimension values.
- If column headers are specified, the number of column headers must equal the number of columns in each record.

External Data Limits

For more information on limits for all supported editions, see External Data Limits in CRM Analytics Limits.

External Data Reference

The InsightsExternalData Object

With the InsightsExternalData object, you can configure and control external data uploads. You can use it to provide metadata, trigger the start of the upload process, check status, and request cancellation and cleanup.

The InsightsExternalData object is used with the InsightsExternalDataPart object, which holds the parts of the data to be uploaded. Together, they provide a programmatic way to upload a large file in parts and trigger a dataflow into a dataset. The first step is to insert a row into the InsightsExternalData object. Data parts are then uploaded to InsightsExternalDataPart objects. The `Mode` field is used to enable faster uploads. The `Action` field of the InsightsExternalData object is updated to start processing and request cancellations. After the `Action` field is updated to request processing, no user edits are allowed on the objects, except to request cancellation.

Note: The standard system fields (`CreatedById`, `CreatedDate`, `LastModifiedById`, `LastModifiedDate`, and `SystemModstamp`) are documented in System Fields in the *Salesforce Object Reference*.

The InsightsExternalData object is available in API version 31 and later.

Supported Calls

`create()`, `delete()`, `describeSObjects()`, `query()`, `retrieve()`, `update()`, `upsert()`

Fields

Field	Details
`Action`	**Type** Picklist **Properties** Create, Filter, Group, Sort, Update **Description** The action to perform on this data. Picklist values are: **`Abort`** **Reserved for future use.** The user no longer wants to upload the data and is requesting that the system stop processing, if possible. **`Delete`** **Reserved for future use.** The user wants to remove uploaded data parts as soon as possible. Implies that an Abort status is queued. **`None`** The user hasn't completed the data upload. This value is the default when the object is created. **`Process`** The user completed the data upload and is requesting that the system process the data.
`CompressedMetadataLength`	**Type** Int

Field	Details
	Properties Create, Filter, Group, Nillable, Sort, Update **Description** The length of the compressed metadata .json file. This field is overwritten when data is uploaded. This system field isn't editable.
`Dataflow`	**Type** String **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** For dataflows that were created in API version 34.0 and later. The unique ID of the dataflow that was used to create the dataset. You can use this field to get the status of the dataflow. This system field isn't editable.
`Description`	**Type** String **Properties** Create, Filter, Nillable, Sort, Update **Description** The description of the dataset that is only used when creating the dataset.
`EdgemartAlias`	**Type** String **Properties** Create, Filter, Group, Sort, Update **Description** The alias of a dataset, which must be unique across an organization. The alias must follow the same guidelines as other field names, except that they can't end with "__c". Can be up to 80 characters. For more information, see *Field Names* in the CRM Analytics External Data Format Developer Guide.
`EdgemartContainer`	**Type** String **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** The name of the app that contains the dataset. • If the name is omitted when you're creating a dataset, the name of the user's private app is used. • If the name is omitted for an existing dataset, the system resolves the app name.

Field	Details
	• If the name is specified for an existing dataset, the name is required to match the name of the current app that contains the dataset. Note: Use the developer name or the ID of the app for the name. To get the developer name or ID, run this query: `SELECT Id,DeveloperName,Name, AccessType,CreatedDate,Type FROM Folder where Type = 'Insights'` For example, the display label of an app is Analytics Cloud Public Datasets, but the developer name is AnalyticsCloudPublicDatasets.
`EdgemartLabel`	**Type** String **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** The display name for the dataset. Can be up to 255 characters.
`FileName`	**Type** String **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** Identifier of the external data file, such as the file name. A unique value isn't required. It can contain only alphanumeric characters and underscores. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. The maximum file name is 255 characters. Note: CRM Analytics doesn't populate this field. You can manually update it via the External Data API.
`Format`	**Type** Picklist **Properties** Create, Filter, Group, Sort, Update **Description** The format of the uploaded data. Picklist values are: **`Csv`** The data is in CSV format. **`Binary`** Reserved for Salesforce internal use.

Field	Details
`isDependentOnLastUpload`	**Type** Boolean **Properties** Create, Defaulted on create, Filter, Group, Sort, Update **Description** **Reserved for future use.** When `false`, indicates that this upload depends on the previous upload to the same dataset name.
`isIndependentParts`	**Type** Boolean **Properties** Create, Defaulted on create, Filter, Group, Sort, Update **Description** **Reserved for future use.** When `true`, indicates that file parts were divided on row boundaries and can be processed independently of each other. The default is `false`.
`LicenseType`	**Type** picklist **Properties** Create, Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort, Update **Description** The license type of the external data file. Possible values are: • `Aqs` (Analytics Query Service) • `Cdp` (Data Cloud) • `DataPipelineQuery` (Data Pipeline Query) • `EinsteinAnalytics` (CRM Analytics) • `IntelligentApps` (Intelligent Apps) • `Sonic` (Salesforce Data Pipelines) The default value is `EinsteinAnalytics`.
`MetaDataLength`	**Type** Int **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** The length of the metadata .json file. This field is overwritten when data is uploaded. This system field isn't editable.
`MetadataJson`	**Type** Blob (Base64-encoded string)

Field	Details
	Properties Create, Nillable, Update **Description** Metadata in JSON format, which describes the structure of the uploaded file.
`Mode`	**Type** picklist **Properties** Create, Filter, Group, Nillable, Restricted picklist, Sort, Update **Description** The upload mode for the data. Possible values are: • `Incremental` • `None` The default value is `None`.
`NotificationEmail`	**Type** String **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** The email address to send notifications to. Can be up to 255 characters and can contain only one email address. Defaults to the current user's email address.
`NotificationSent`	**Type** Picklist **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** Indicates when to send notifications about the upload. Picklist values are: **`Always`** Always send notifications. **`Never`** Never send notifications. **`Failures`** Send notifications if the upload process failed. **`Warnings`** Send notifications if warnings occurred during the upload. **`Success`** Send notifications if the upload is successful.

Field	Details
	`SuccessFailures` Send notifications if the upload is successful or if the process failed. **`WarningsFailures`** Send notifications if warnings or failures occurred during the upload. **`WarningsSuccess`** Send notifications if the upload is successful or if warnings occurred during the upload. The default value is 'WarningsFailures'.
`Operation`	**Type** Picklist **Properties** Create, Filter, Group, Sort, Update **Description** Indicates which operation to use when you're loading data into the dataset. Picklist values are: **`Append`** Append all data to the dataset. Creates a dataset if it doesn't exist. Note: If the dataset or rows contain a unique identifier, the append operation isn't allowed. **`Delete`** Delete the rows from the dataset. The rows to delete must contain one (and only one) field with a unique identifier. **`Overwrite`** Create a dataset with the given data, and replace the dataset if it exists. **`Upsert`** Insert or update rows in the dataset. Creates a dataset if it doesn't exist. The rows to upsert must contain one (and only one) field with a unique identifier. For more information about unique identifiers, see *isUniqueId* in the CRM Analytics External Data Format Developer Guide. Note: A metadata JSON file is required for the append, upsert, and delete operations. The data and metadata for the append and upsert operations must match the dataset on which the operation is happening. (All columns, dimensions, and measures must match exactly.) The metadata for the delete operation must be a subset of the dataset columns.
`Status`	**Type** Picklist **Properties** Create, Filter, Group, Sort, Update **Description** The status of this data upload. The initial value is null. Picklist values are:

Field	Details
	`Completed` The data upload job was completed successfully. Data parts are retained for 7 days after completion. **`CompletedWithWarnings`** The data upload job completed, but contains warnings. Data parts are retained for 7 days after completion. **`Failed`** The data upload job failed. Data parts are retained for 7 days after failure. **`InProgress`** The data upload job is in progress. **`New`** The data upload job has been created. **`NotProcessed`** The data upload job was aborted on user request. Data parts have been removed. **`Queued`** The data upload job has been scheduled. This system field isn't editable. The default value is `New`.
`StatusMessage`	**Type** String **Properties** Create, Nillable, Update **Description** The reason for the file upload failed or has warnings. This system field isn't editable.
`SubmittedDate`	**Type** String **Properties** Create, Filter, Nillable, Sort, Update **Description** The time when the upload was submitted or set to Process. This system field isn't editable.
`Target`	**Type** picklist **Properties** Create, Filter, Group, Nillable, Restricted picklist, Sort, Update **Description** The target for the external data. Valid values are `Dataset`.

The InsightsExternalDataPart Object

The InsightsExternalDataPart object enables you to upload an external data file that has been split into parts.

The InsightsExternalDataPart object works with the InsightsExternalData object. After you insert a row into the InsightsExternalData object, you can create part objects to split up your data into parts. If your initial data file is larger than 10 MB, split your file into parts that are smaller than 10 MB.

Note: The standard system fields (`CreatedById`, `CreatedDate`, `LastModifiedById`, `LastModifiedDate`, and `SystemModstamp`) are documented in System Fields in the *Salesforce Object Reference*.

The InsightsExternalDataPart object is available in API version 31 and later.

Supported Calls

`create()`, `delete()`, `describeSObjects()`, `query()`, `retrieve()`, `update()`, `upsert()`

Fields

Field	Details
`CompressedDataLength`	**Type** Int **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** The length of the compressed data. This field is overwritten when data is uploaded.
`DataFile`	**Type** Blob (Base64-encoded string) **Properties** Create, Nillable, Update **Description** The data bytes. Parts are required to be smaller than 10 MB. For data greater than 10 MB, compress the file and then split it into parts. Only the gzip format is supported.
`DataLength`	**Type** Int **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** The length of the data. This field is overwritten when data is uploaded.
`InsightsExternalDataId`	**Type** String

Field	Details
	Properties Create, Filter, Group, Sort **Description** The ID of the InsightsExternalData object that this part belongs to.
`PartNumber`	**Type** Int **Properties** Create, Filter, Group, Sort, Update **Description** The part number. Part numbers are required to be in a contiguous sequence, starting with 1. (For example, 1, 2, 3, etc.)

External Data Metadata Format Reference

The metadata describes the structure of external data files. The metadata file is in JSON format. The .json file consists of 3 main sections: file format, object information, and field information. Include all required fields when you create a record, but you can leave out optional fields.

The File Format Section

The file format section of the metadata file specifies information about the format of the data file, including the character set and delimiter character.

Field Name	Type	Required?	Description
`charsetName`	String	No	The character set of the .csv file. If this field is included, it must be set to UTF-8. Example: `"charsetName": "UTF-8"`
`fieldsDelimitedBy`	String	No	The character that separates field values in the .csv file. Any single character is supported. Example: `"fieldsDelimitedBy": ","`
`fieldsEnclosedBy`	String	No	The character that can be used to enclose fields in the .csv file. Any single character that isn't the `fieldsDelimitedBy` value is supported. If a double quote is used within a field, escape it by preceding it with another double quote. Example: `"fieldsEnclosedBy": "\""`

Field Name	Type	Required?	Description

`fieldsEscapedBy`	String	No	The character used to escape, or bypass, the creation of a row of data based on detecting the `fieldsDelimitedBy` value. The default is a double quote, ", which would surround the data to escape.
			Example:
			`"fieldsEscapedBy": "\\"`
			Example:
			This is an alternate escaping option to `fieldsEnclosedBy`. To illustrate, if `fieldsDelimitedBy` is comma, `fieldsEnclosedBy` is double quote, and `fieldsEscapedBy` is backslash, these two lines would result in the same thing:
			`col1,"col,with,commas",col3`
			`col1,col\,with\,commas,col3`

`linesTerminatedBy`	String	No	*Deprecated. Do not use.*
`numberOfLinesToIgnore`	Number	No	The number of lines for the parser to ignore. (Allows you to specify a header.) • When the .csv file doesn't have a header, set to 0. • When the .csv file has a header, set to 1. Example: `"numberOfLinesToIgnore": 1`

The Objects Section

The objects section of the metadata file specifies information about the top-level database object, including object-level security information, display name, and API name.

Note: The metadata file can contain only 1 object definition.

Field Name	Type	Required?	Description
`rowLevelSecurityFilter`	String	No	The predicate that's used to apply row-level security on the dataset. When entering the predicate in the metadata file JSON, escape double quotes around string values. Example: `"rowLevelSecurityFilter": "'OwnerId' == \"$User.Id\""` For more information about creating the predicate, see the *Analytics Security Implementation Guide*.

Field Name	Type	Required?	Description
`connector`	String	Yes	The string that uniquely identifies the client application. Example: `"connector": "AcmeCSVConnector"`
`description`	String	No	The description of the object. Must be less than 1,000 characters. Example: `"description": "The SalesData object tracks basic sales data."`
`fullyQualifiedName`	String	Yes	The full path that uniquely identifies the record. Must be less than 1,000 characters. Example: `"fullyQualifiedName": "CRM.SalesData"` For information on creating valid field names, see Field Name Restrictions in this topic.
`label`	String	Yes	The display name for the object. Can be up to 40 characters. Example: `"label": "Sales Data"`
`name`	String	Yes	The unique API name for the object. Can be up to 255 characters. Example: `"name": "SalesData"` For information on creating valid field names, see Field Name Restrictions in this topic.
`fields`	Array	Yes	The array of fields for this object.

The Fields Section

The fields section of the metadata file specifies information about each field in the record, including data type and formatting information.

Note: The fields must be in the same order as the CSV columns are in.

Field Name	Type	Required?	Description
`fullyQualifiedName`	String	Yes	The full path that uniquely identifies the field (object.field). Must be less than 1,000 characters. Example: `"fullyQualifiedName": "SalesData.Amount"`

Field Name	Type	Required?	Description
			For information on creating valid field names, see Field Name Restrictions in this topic.
`label`	String	Yes	The display name for the field. Can be up to 255 characters. Example: `"label": "Opportunity Amount"`
`name`	String	Yes	The unique API name for the field. Can be up to 255 characters. Example: `"name": "Amount"` For information on creating valid field names, see Field Name Restrictions in this topic.
`description`	String	No	The description of the field. Must be less than 1,000 characters. Example: `"description": "The Amount field contains the opportunity amount."`
`isSystemField`	Boolean	No	Indicates whether this field is a system field to be excluded from query results. Example: `"isSystemField": false`
`type`	String	Yes	The type of the field. Can be Text, Numeric, or Date. Example: `"type": "Numeric"`
`defaultValue`	String	No	The default value of the field, if any. All numeric types require a default value.
`isUniqueId`	Boolean	No	Indicates whether this field is the primary key for the record. This field is required for incremental extract. Only 1 field can be set to be the unique ID. Only text fields can be unique IDs. Numeric, date, and multivalue fields can't be unique IDs. Example: `"isUniqueId": false`
`isMultiValue`	Boolean	No	For text fields only. Indicates whether the field has multiple values. Applies only to Text fields. Example:

Field Name	Type	Required?	Description
			`"isMultiValue": false`
`multiValueSeparator`	String	No	For text fields only. The character that separates multiple values. The default is ";". • If `isMultiValue` equals true, specify a value. • If `isMultiValue` equals false, this field can be set to null. Example: `"multiValueSeparator": ";"`

`format`	String	Yes (for Date values only)	The format of the date or numeric value. See also: Date Formats and Numeric Formats.
			Example:
			`"format": "dd-MM-yy HH:mm:ss" (Date)`
			Example:
			`"format": "$#,##0.00" (Numeric)`

`precision`	Number	Yes (for Numeric values)	The maximum number of digits in a numeric value, or the length of a text value. For numeric values: Includes all numbers to the left and to the right of the decimal point (but excludes the decimal point character). Value can be up to 18. For text values: Value defaults to 255 characters, but can be up to 32,000 characters. Example: `"precision": 10`
`scale`	Number	Yes (for Numeric values)	The number of digits to the right of the decimal point in a numeric value. Must be less than the precision value. Example: `"scale": 2`
`canTruncateValue`	Boolean	No	For text fields only. Indicates whether to truncate a value when the value exceeds the precision. The default is true. • If true, truncates the value. • If false, the row is rejected. Example: `"canTruncateValue": true`
`currencySymbol`	String	No	For numeric fields only. The character that signifies the value's currency. Example: `"currencySymbol": "£"`

Field Name	Type	Required?	Description
`decimalSeparator`	String	No	For numeric fields only. The character that separates digits in a decimal number. Can be used to handle international number formats where the decimal separator is ",". The default is ".". Example: `"decimalSeparator": ","`
`fiscalMonthOffset`	Number	No	For date fields only. The difference, in months, between the fiscal year and the calendar year. For example, if the fiscal year starts in January, the offset is 0. If the fiscal year starts in October, the offset is 9. Example: `"fiscalMonthOffset": 9` This attribute also controls whether Analytics generates fiscal date fields. To generate fiscal date fields, set `fiscalMonthOffset` to a value other than 0. See also Date Handling in Datasets.
`groupSeparator`	String	No	For numeric fields only. The character that separates digit groups in a number. Can be used to handle international number formats where the group separator is ".". The default is ",". Example: `"groupSeparator": "."`
`isYearEndFiscalYear`	Boolean	No	For date fields only. Indicates whether the fiscal year is the year in which the fiscal year ends or begins. Because the fiscal year can start in one calendar year and end in another, specify which year to use for the fiscal year. • If true, then the fiscal year is the year in which the fiscal year ends. The default is true. • If false, then the fiscal year is the year in which the fiscal year begins. Example: `"isYearEndFiscalYear": true` This field is relevant only when `fiscalMonthOffset` is greater than 0. See also Date Handling in Datasets.
`firstDayOfWeek`	Number	No	For date fields only. The first day of the week for the calendar year and, if applicable, fiscal year. Use 0 to set the first day to be Sunday, 1 to set the first day to be Monday, and so on. Use -1 to set the first day to be January 1. The default is -1. Example: `"firstDayOfWeek": 0` See also Date Handling in Datasets.

Field Name	Type	Required?	Description
`isSkipped`	Boolean	No	Indicates whether to skip the field when the data is uploaded. Example: `"isSkipped": true`

Field Name Restrictions

Field names in the .csv file and the metadata file:

- Can contain only alphanumeric and underscore characters
- Must begin with a letter
- Can't end with an underscore
- Can't contain 2 consecutive underscore characters, except when ending with "__c" (case-sensitive)
- Must be unique across all fields of the object

Note: We recommend that field names in dataset use no more than 40 characters. Long field names increase the likelihood of exceeding character limits when you augment dataset, because names are appended.

Numeric Formats

An example of a typical numeric value is $1,000,000.99, which is represented as $#,##0.00 in the `format` field. You're required to specify the precision and scale of the number. The format is specified by using the following symbols:

Symbol	Meaning
0	One digit. Use to add leading or trailing 0s, like `#,###.00` for $56,375.00.
#	Adds zero or one digit
.	Default symbol used as the decimal separator. Use the `decimalSeparator` field to set the decimal separator to a different symbol.
-	Minus sign
,	Grouping separator. Use the `groupSeparator` field to set the group separator to a different symbol.
$	Currency sign. Use the `currencySymbol` field to set the currency indicator to a different symbol.

Note: The format for numeric values when displayed in the UI defaults to No Format. Existing formatting is removed. For data ingestion, numeric values can't contain any formatting (such as currency symbols or grouping separators). For example, $1,000.00 isn't a valid numeric value; the correct value is 1000.00.

Valid characters when defining a numeric format are:

- 0
- #
- ,

- -
- ,
- $

Date Formats

For Date fields, specify the format of the date by using one of the following supported formats. Dates must match the format exactly and can't have any extra text. For example, if the date format is "MM-dd-yyyy hh:mm:ss" and the value is "12-31-2015 12:00:00.0000", the upload fails because the value has extra milliseconds.

Note: The date formats listed here are the two-digit versions for date fields that use leading zeros; for example,`03/06/14 09:01:06 AM`. If a date field doesn't have leading zeros, use the one-digit version of the format. For example, use the format `M/d/yy h:m:s a` for date values such as **3/6/14 9:1:26 AM**. If you use a two-digit format for a field, rows containing values with one-digit date parts will fail.

The timestamp part of each date format is optional.

Format	Sample Value
yyyy-MM-dd'T'HH:mm:ss.SSS'Z'	2014-04-29T16:53:34.000Z
yy-MM-dd'T'HH:mm:ss.SSS'Z'	14-04-29T16:53:34.000Z
yyyy-MM-dd'T'HH:mm:ss'Z'	2014-04-29T16:53:34Z
yy-MM-dd'T'HH:mm:ss'Z'	14-04-29T16:53:34Z
yyyy-MM-dd HH:mm:ss	2014-06-03 11:31:45
yy-MM-dd HH:mm:ss	14-06-03 11:31:45
dd.MM.yyyy HH:mm:ss	03.06.2014 11:31:45
dd.MM.yy HH:mm:ss	03.06.14 11:31:45
dd/MM/yyyy HH:mm:ss	03/06/2014 11:31:45
dd/MM/yy HH:mm:ss	03/06/14 11:31:45
dd/MM/yyyy hh:mm:ss a	03/06/2014 11:31:45 AM
dd/MM/yy hh:mm:ss a	03/06/14 11:31:45 AM
dd-MM-yyyy HH:mm:ss	03-06-2014 11:31:45
dd-MM-yy HH:mm:ss	03-06-14 11:31:45
dd-MM-yyyy hh:mm:ss a	03-06-2014 11:31:45 AM
dd-MM-yy hh:mm:ss a	03-06-14 11:31:45 AM
MM/dd/yyyy hh:mm:ss a	06/03/2014 11:31:45 AM
MM/dd/yy hh:mm:ss a	06/03/14 11:31:45 AM
MM-dd-yyyy hh:mm:ss a	06-03-2014 11:31:45 AM

Format	Sample Value
MM-dd-yy hh:mm:ss a	06-03-14 11:31:45 AM
HH:mm:ss dd/MM/yyyy	11:31:45 03/06/2014
HH:mm:ss dd/MM/yy	11:31:45 03/06/14

These formats use the following symbols:

Symbol	Meaning
yyyy or yy	Four-digit year (yyyy) or two-digit year (yy)
MM	Two-digit month (01–12)
M	One-digit month when month less than 10 (1–12)
dd	Two-digit day (01–31)
d	One-digit day when day less than 10 (1–31)
'T'	A separator that indicates that time of day follows
HH	Two-digit hour (00–23)
H	One-digit hour when hour less than 10 (0–23)
mm	Two-digit minute (00–59)
m	One-digit minute when minute less than 10 (0–59)
ss	Two-digit second (00–59)
s	One-digit second when second less than 10 (0–59)
SSS	Optional three-digit milliseconds (000–999)
'Z'	The reference UTC time zone

RELEASE NOTES

Use the Salesforce Release Notes to learn about the most recent updates and changes to the Analytics External Data API.

For a list of all current developer changes, including Analytics External Data API, see CRM Analytics in the Salesforce Release Notes.

Note: If the Analytics Development section in the Salesforce Release Notes isn't present, there aren't any updates for that release.