
Wave Analytics External Data API Developer Guide

Salesforce, Winter '17

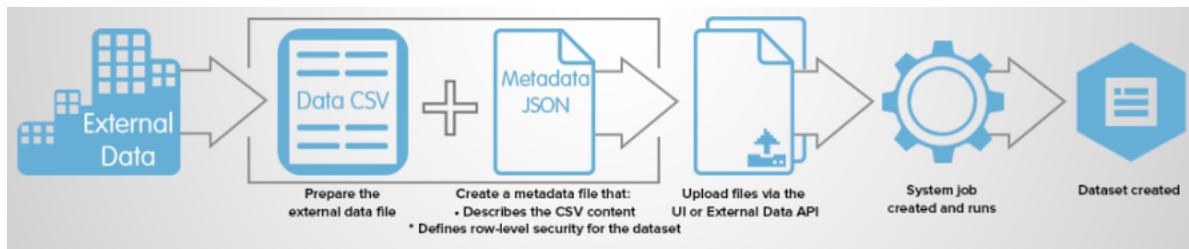


CONTENTS

- [External Data](#) 1
- Load External Data into a Dataset 1
- External Data API Reference 4

EXTERNAL DATA

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



The External Data API enables you to upload external data files to Wave 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.

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 Wave Analytics limits.

The following limits apply to all supported editions. Contact salesforce.com to extend the limits.

External Data Limits

Limit	Value
Maximum file size per external data upload	If you use the External Data API: 40 GB. If you use the user interface: 512 MB.
Maximum file size for all external data uploads in a rolling 24-hour period	50 GB
Maximum number of external data files that can be uploaded in a rolling 24-hour period	50

Limit	Value
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

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 [Wave Analytics External Data Format Reference](#).

Connect to Salesforce

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



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

To load external data into Wave Analytics, first connect to your Salesforce organization. Use the PartnerConnection object to log in to your organization, as shown in the following example. You'll 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 Force.com Web Services Connector](#) on the Salesforce Developers website. For more information about user authentication, see [Security and the API](#) in the *SOAP API Developer's 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 the REST API, you'll need to make this conversion yourself.

For detailed information about the `InsightsExternalData` object, see [InsightsExternalData](#).

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();
soobj.setType("InsightsExternalData");
soobj.setField("Action","Process");
soobj.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.

External Data API Reference

The InsightsExternalData Object

The `InsightsExternalData` object enables you to configure and control external data uploads. You can use it to provide metadata, trigger the start of the upload process, check status, and request cancelation 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 `InsightsExternalData` objects. The `Action` field of the `InsightsExternalData` object is updated to start processing and request cancelations. After the `Action` field is updated to request processing, no user edits are allowed on the objects, except to request cancelation.






Note: The standard system fields (`CreatedById`, `CreatedDate`, `LastModifiedById`, `LastModifiedDate`, and `SystemModstamp`) are documented in "System Fields" in the *SOAP API Developer's Guide*.

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

Fields

Field	Details
<code>Id</code>	<p>Type String</p> <p>Properties Defaulted on create, Filter, Group, Sort</p> <p>Description The unique ID that the system generates for new jobs. This system field is not editable.</p>


Field	Details
EdgemartAlias	<p>Type String</p> <p>Properties Create, Filter, Group, Sort, Update</p> <p>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 <i>Field Names</i> in the External Data Format Reference.</p>
EdgemartContainer	<p>Type String</p> <p>Properties Create, Filter, Group, Nillable, Sort, Update</p> <p>Description The name of the app that contains the dataset.</p> <ul style="list-style-type: none"> • 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. • 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. <p> Note: Use the developer name or the ID of the app for the name. To get the developer name or ID, run the following query:</p> <pre>SELECT Id, DeveloperName, Name, AccessType, CreatedDate, Type FROM Folder where Type = 'Insights'</pre> <p>For example, the display label of an app is Analytics Cloud Public Datasets, but the developer name is AnalyticsCloudPublicDatasets.</p>
MetadataJson	<p>Type Blob (Base64-encoded string)</p> <p>Properties Create, Nillable, Update</p> <p>Description Metadata in JSON format, which describes the structure of the uploaded file.</p>
Format	<p>Type Picklist</p> <p>Properties Create, Filter, Group, Sort, Update</p>

Field	Details
	<p>Description</p> <p>The format of the uploaded data. Picklist values are:</p> <p>Csv</p> <p>The data is in CSV format.</p> <p>Binary</p> <p>Reserved for Salesforce internal use.</p>
Operation	<p>Type</p> <p>Picklist</p> <p>Properties</p> <p>Create, Filter, Group, Sort, Update</p> <p>Description</p> <p>Indicates which operation to use when you're loading data into the dataset. Picklist values are:</p> <p>Append</p> <p>Append all data to the dataset. Creates a dataset if it doesn't exist.</p> <p> Note: If the dataset or rows contain a unique identifier, the append operation is not allowed.</p> <p>Delete</p> <p>Delete the rows from the dataset. The rows to delete must contain one (and only one) field with a unique identifier.</p> <p>Overwrite</p> <p>Create a dataset with the given data, and replace the dataset if it exists.</p> <p>Upsert</p> <p>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 <i>isUniqueId</i> in the External Data Format Reference.</p> <p> 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.</p>
Status	<p>Type</p> <p>Picklist</p> <p>Properties</p> <p>Create, Filter, Group, Sort, Update</p> <p>Description</p> <p>The status of this data upload. The initial value is null. Picklist values are:</p> <p>Completed</p> <p>The data upload job was completed successfully. Data parts are retained for seven days after completion.</p>

Field	Details
	<p>CompletedWithWarnings The data upload job completed, but contains warnings. Data parts are retained for seven days after completion.</p> <p>Failed The data upload job failed. Data parts are retained for seven days after failure.</p> <p>InProgress The data upload job is in progress.</p> <p>New The data upload job has been created.</p> <p>NotProcessed The data upload job was aborted on user request. Data parts have been removed.</p> <p>Queued The data upload job has been scheduled. This system field is not editable.</p>
Action	<p>Type Picklist</p> <p>Properties Create, Filter, Group, Sort, Update</p> <p>Description The action to perform on this data. Picklist values are:</p> <p>Abort Reserved for future use. The user no longer wants to upload the data and is requesting that the system stop processing, if possible.</p> <p>Delete Reserved for future use. The user wants to remove uploaded data parts as soon as possible. Implies that an Abort status is queued.</p> <p>None The user has not completed the data upload. This value is the default when the object is created.</p> <p>Process The user has completed the data upload and is requesting that the system process the data.</p>
isIndependentParts	<p>Type Boolean</p> <p>Properties Create, Defaulted on create, Filter, Group, Sort, Update</p> <p>Description Reserved for future use. When <code>true</code>, indicates that file parts have been divided on row boundaries and can be processed independently of each other. The default is <code>false</code>.</p>

Field	Details
isDependentOnLastUpload	<p>Type Boolean</p> <p>Properties Create, Defaulted on create, Filter, Group, Sort, Update</p> <p>Description Reserved for future use. When <code>false</code>, indicates that this upload depends on the previous upload to the same dataset name.</p>
MetaDataLength	<p>Type Int</p> <p>Properties Create, Filter, Group, Nillable, Sort, Update</p> <p>Description The length of the metadata .json file. This field is overwritten when data is uploaded. This system field is not editable.</p>
CompressedMetaDataLength	<p>Type Int</p> <p>Properties Create, Filter, Group, Nillable, Sort, Update</p> <p>Description The length of the compressed metadata .json file. This field is overwritten when data is uploaded. This system field is not editable.</p>
NotificationSent	<p>Type Picklist</p> <p>Properties Create, Filter, Group, Nillable, Sort, Update</p> <p>Description Indicates when to send notifications about the upload. Picklist values are:</p> <p>Always Always send notifications.</p> <p>Never Never send notifications.</p> <p>Failures Send notifications if the upload process failed. The API name is Failures.</p> <p>Warnings Send notifications if warnings or errors occurred during the upload. The API name is Warnings.</p>


Field	Details
NotificationEmail	<p>Type String</p> <p>Properties Create, Filter, Group, Nillable, Sort, Update</p> <p>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.</p>
EdgemartLabel	<p>Type String</p> <p>Properties Create, Filter, Group, Nillable, Sort, Update</p> <p>Description The display name for the dataset. Can be up to 255 characters.</p>
IsDeleted	<p>Type Boolean</p> <p>Properties Defaulted on Create, Filter, Group, Sort</p> <p>Description Indicates whether the object has been moved to the Recycle Bin (<code>true</code>) or not (<code>false</code>). This system field is not editable.</p>
Dataflow	<p>Type String</p> <p>Properties Create, Filter, Group, Nillable, Sort, Update</p> <p>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 is not editable.</p>
WorkflowId	<p>Type String</p> <p>Properties Create, Filter, Group, Nillable, Sort, Update</p> <p>Description Deprecated in API version 34.0. Use the <code>Dataflow</code> attribute instead.</p>
SubmittedDate	<p>Type String</p>

Field	Details
	<p>Properties Create, Filter, Nillable, Sort, Update</p> <p>Description The time when the upload was submitted or set to Process. This system field is not editable.</p>
FileName	<p>Type String</p> <p>Properties Create, Filter, Group, Nillable, Sort, Update</p> <p>Description Identifier of the external data file, such as the file name. The value does not have to be unique. 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.</p> <p> Note: Wave Analytics doesn't populate this field. You can manually update it via the External Data API.</p>
Description	<p>Type String</p> <p>Properties Create, Filter, Nillable, Sort, Update</p> <p>Description The description of the dataset that is only used when creating the dataset.</p>
StatusMessage	<p>Type String</p> <p>Properties Create, Nillable, Update</p> <p>Description The reason for the file upload failed or has warnings. This system field is not editable.</p>

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 (CreatedBy, CreatedDate, LastModifiedBy, LastModifiedDate, and SystemModstamp) are documented in “System Fields” in the *SOAP API Developer's Guide*.

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

Fields

Field	Details
Id	<p>Type String</p> <p>Properties Defaulted on create, Filter, Group, Sort</p> <p>Description The unique ID of the part.</p>
PartNumber	<p>Type Int</p> <p>Properties Create, Filter, Group, Sort, Update</p> <p>Description The part number. Part numbers are required to be in a contiguous sequence, starting with 1. (For example, 1, 2, 3, etc.)</p>
InsightsExternalDataId	<p>Type String</p> <p>Properties Create, Filter, Group, Sort</p> <p>Description The ID of the InsightsExternalData object that this part belongs to.</p>
DataFile	<p>Type Blob (Base64-encoded string)</p> <p>Properties Create, Nillable, Update</p> <p>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.</p>
DataLength	<p>Type Int</p> <p>Properties Create, Filter, Group, Nillable, Sort, Update</p> <p>Description The length of the data. This field is overwritten when data is uploaded.</p>

Field	Details
CompressedDataLength	<p>Type Int</p> <p>Properties Create, Filter, Group, Nillable, Sort, Update</p> <p>Description The length of the compressed data. This field is overwritten when data is uploaded.</p>
IsDeleted	<p>Type Boolean</p> <p>Properties Defaulted on Create, Filter, Group, Sort</p> <p>Description Indicates whether the object has been moved to the Recycle Bin (<code>true</code>) or not (<code>false</code>).</p>