TIBCO ActiveMatrix BusinessWorks™ Plug-in for Salesforce.com

User's Guide

Version 6.9.2

November 2021

Contents

Plug-In Overview

You can leverage the benefits of the Salesforce.com server without any coding by using TIBCO ActiveMatrix BusinessWorks™ Plug-in for Salesforce.com.

With this plug-in, you can configure a connection to the Salesforce.com server and use activities to integrate with the Salesforce.com database.

ActiveMatrix BusinessWorks™ Plug-in for Salesforce.com provides the following common functions for integrating with the Salesforce.com server:

- Easy-to-configure connection configuration shared resource. You can enable mutual authentication by providing SSL Client Configuration.

  This shared resource can be reused for different activities and processes.

- Get Session activity

  This activity provides a simple interface and hides all details of complex session management tasks, such as login, session expiration, and so on.

- Metadata Management

  You can build metadata at design time and store the metadata as XML Schema Definition (XSD) schemas, which in turn is used at run time to perform runtime type-checking.

- High level data calls

  - Create All, Query All, Update All, Upsert All, Retrieve All, and Delete All from the SOAP-based Salesforce palette.

  - Bulk Operation, Bulk Query, Check Status, Get Result, Get Subset Result, and Close Job from the REST-based Salesforce Bulk API palette.

  - Bulk Ingest Operation, Bulk Ingest Query, Check Ingest Job Status, Get Ingest Job Result, and Close Ingest Job from the REST-based Salesforce Bulk API 2.0 palette.

  - Composite Batch, Composite Tree, and Composite Dependent from the REST-based Composite API palette.

  Each of these activities supports the mechanism of built-in error handling and logging.

- Dynamic Web Services Description Language (WSDL) switching

  You can work with different versions of Salesforce.com Partner and Enterprise WSDLs by using this function.

- Salesforce.com Streaming Listener

  The listener is implemented as a process starter using the Salesforce Subscriber activity on the SOAP based Salesforce palette. The types of subscriber are Push Topic, Platform Event, and Change Data Capture. You can store the last processed replay ID in a database and replay the messages from that replay ID onward. You can apply filters on Push Topic subscription.

- Outbound Message Listener

  You can receive notifications from the Salesforce.com server by using this function.

- Relationship Query

  You can use this function to query multiple sObjects at one time in the Salesforce.com database. A relationship is created between those sObjects. Salesforce Object Query Language (SOQL) provides the syntax to support relationship queries. Parent-to-children and child-to-parent are the two possible relationships between sObjects.

- Salesforce Query Builders

– Salesforce Query Builder provides you with a wizard to construct an SOQL query.

– Salesforce Search Query Builder provides you with a wizard to construct an SOSL query.

Getting Started

A typical workflow using the plug-in to achieve different goals includes creating a project, designing a process, and deploying the application.

By using TIBCO ActiveMatrix BusinessWorks, you can create services and integrate applications, and deploy them at run time. It uses the Eclipse graphical user interface (GUI) for defining business processes and the process engine to execute them.

Complete the following tasks to design a process and deploy it at run time:

1. Creating a Project
2. Importing a WSDL
3. Creating a Salesforce Connection
4. Configuring a Process
5. Debugging and Running a Process
6. Deploying an Application

Creating a Project

The first task of using the plug-in is creating a project. After creating a project, you can add resources and processes.

An Eclipse project is an application module configured for TIBCO ActiveMatrix BusinessWorks. An application module is the smallest unit of resources that is named, versioned, and packaged as part of an application.

Procedure

1. Start TIBCO Business Studio™ for BusinessWorks in one of the following ways:

   - Microsoft Windows: click **Start** > **All Programs** > **TIBCO** > *TIBCO_HOME* > **TIBCO Business Studio** *version_number* > **Studio for Designers**.
   - Mac OS and Linux: run the TIBCO Business Studio for BusinessWorks executable file located in the `TIBCO_HOME`/studio/`version_number`/eclipse directory.

2. From the menu, click **File** > **New** > **BusinessWorks Resources** to open the BusinessWorks Resource wizard.

3. In the Select a wizard dialog box, click **BusinessWorks Application Module** and click **Next** to open the New BusinessWorks Application Module wizard.

4. In the Project dialog box, configure the project that you want to create:
   a) In the **Project name** field, enter a project name.
   b) By default, the created project is located in the workspace current in use. If you do not want to use the default location for the project, clear the **Use default location** check box and click **Browse** to select a new location.
   c) Use the default version of the application module, or enter a new version in the **Version** field.
   d) Keep the **Create empty process** and **Create Application** check boxes selected to automatically create an empty process and an application when creating the project.
   e) Select the **Use Java configuration** check box if you want to create a Java module.

      A Java module provides the Java tooling capabilities.
   f) Click **Finish** to create the project.

Result

A project and an application are created and displayed in the Project Explorer view. The Process editor is displayed automatically.
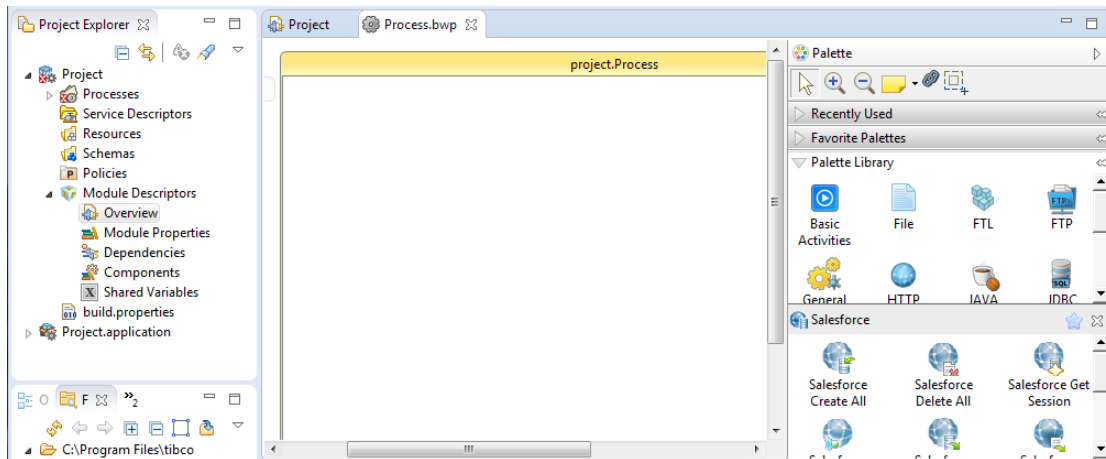
Importing a WSDL

The Web Services Description Language (WSDL) is an XML-based language that can be used to describe the services a business offers and to provide a way for individuals and other businesses to access those services electronically.

Procedure

1. Start TIBCO Business Studio.

2. In the Project Explorer view, right-click the application module and click **BW Salesforce Tools** > **Select Wsdl** from the pop-up menu.

3. Select a WSDL from the **Salesforce WSDL** list, and then click **OK**.

   The default WSDL value is `Partner WSDL`. If you select **Other WSDLs**, you have to locate the corresponding WSDL file in your computer.

4. Click **OK** in the Success window.

   > This manual uses `Partner 47.0 WSDL` to describe the Salesforce activities. For each activity, the fields under the **Input**, **Output**, and **Fault** tabs are generated from the `Partner 47.0` `WSDL`. If you select a WSDL other than this WSDL in a project, the **Input**, **Output**, and **Fault** fields might be different from the fields explained in the Salesforce palette.

What to do next

After selecting a WSDL, you can create a Salesforce connection as described in Create a Salesforce Connection.

Creating a Salesforce Connection Shared Resource

To use the plug-in activities, you must create a Salesforce connection using the Salesforce shared resource. The Salesforce shared resource describes the connection between the plug-in and the Salesforce.com server.

Prerequisites

The Salesforce Connection shared resource is available at the **Resources** level. Before creating a shared Salesforce resource, ensure that you have created a project as described in Creating a Project.

Procedure

1. Expand the created project in the Project Explorer view.

2. Right-click the **Resources** folder and click **New** > **Salesforce Connection**.

3. The resource folder, package name, and resource name of the Salesforce connection are provided by default. If you do not want to use the default configurations, change them accordingly. Click **Finish** to open Salesforce Connection Editor.

4. Configure the related fields in the Salesforce Connection Editor.
   See Salesforce Connection for more details.

5. Click **Test Connection** to validate the connection.

What to do next

After creating the Salesforce connection, fetch the Salesforce metadata from the Salesforce.com server when selecting a Partner WSDL for your project. For more information, see Getting Metadata. The metadata is the key part of the database operations.

Configuring a Process

After creating a project, an empty process is created. You can add activities to the empty process to complete a task.

Procedure

1. Select an activity from the Palette view and drop it in the Process editor.
   For example, select the Timer activity from the **General Activities** palette and drop it in the Process editor.
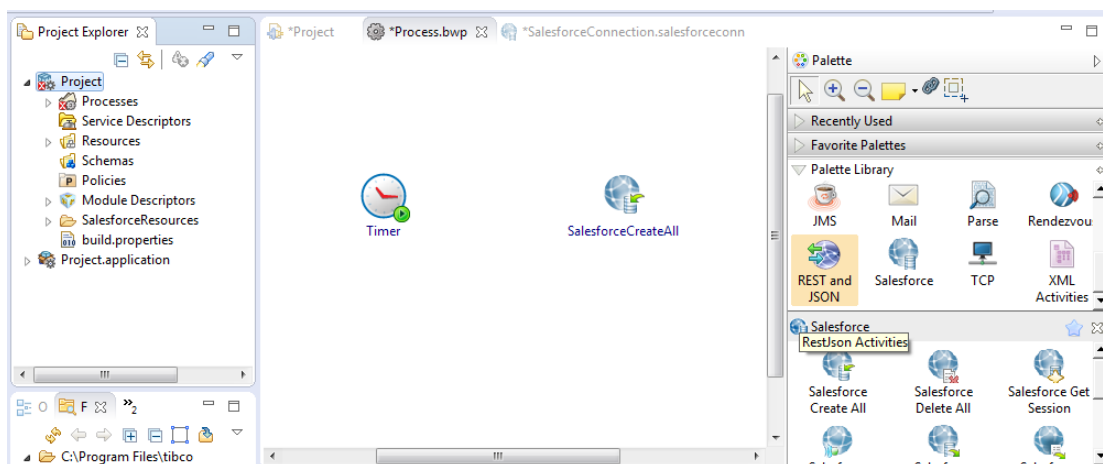
2. Click the **Drag to create a transition between Activities** icon 🔗 to create links between the activities.

3. Configure the added activities.

> 📝 A Salesforce shared resource is required when configuring the Salesforce activities. See Creating a Salesforce Connection for more details on how to create a Salesforce connection.

4. Click **File** > **Save** to save the process.

Debugging and Running a Process

Debug the application you have configured to ensure that the application configuration is correct.

Procedure

1. Open the process you have configured in TIBCO Business Studio.

2. On the toolbar, click **Run** > **Debug Configurations**.

3. Click **BusinessWorks Application** > **BWApplication** in the left panel.

4. Ensure only the application you want to debug and run is selected on the **Applications** tab in the right panel.

5. Click **Debug**.
   TIBCO Business Studio changes to the Debug perspective. Logs are displayed in the Console view.

Deploying an Application

You can manage TIBCO ActiveMatrix BusinessWorks applications by using TIBCO® Enterprise Administrator after deploying the applications.

Prerequisites

An EAR file must be generated before deploying an application. See Generating an EAR File for details about generating an EAR file.

Procedure

1. Build an EAR file.

2. Deploy the EAR file.

   You can deploy an application EAR file from TIBCO Enterprise Administrator, or by using the command-line mode with the **bwadmin** utility. For more information about deploying an application, see *TIBCO ActiveMatrix BusinessWorks Administration*.

3. Configure the application profile.

4. Start the application.

Generating an EAR File

Application archives are the enterprise archive (EAR) files that are created in TIBCO Business Studio. An EAR file is required when deploying an application.

Prerequisites

An application project has been created, as explained in Creating a Project.

Procedure

1. Go to the File Explorer view and click the **Open Directory to Browse** icon .

2. Select the folder where you want to generate the EAR file and click **OK**.
   The new folder is displayed in the File Explorer view.

3. Drag the application from the Project Explorer view to the new folder in the File Explorer view.

   The EAR file is generated with the name `<application>_<version>`.ear. The *<version>* starts from 1.0.0 and increases as more versions occur.

Salesforce Tools

You can use Salesforce tools to manage data on the Salesforce.com server.

It includes the following tools:

- Selecting WSDL
- Getting Metadata
- Refreshing Metadata

Selecting WSDL

The Web Services Description Language (WSDL) is an XML-based language that can be used to describe the services a business offers, and to provide a way for individuals and other businesses to access those services electronically.

Import a WSDL

When working with the plug-in, you must import a WSDL for your project after creating a project. For more information, see Import a WSDL and Creating a Project.

Default WSDL

In the plug-in, all available WSDLs and the `SalesforceWSDLs.xml` file that defines the WSDLs shown in the **Select WSDL** list are located in the *TIBCO_HOME*`\bw\palettes\salesforce\`*version_number*`\wsdls` directory.

The default WSDL is defined in the first `wsdl` node in the `SalesforceWSDLs.xml` file, as shown in the following figure.

```
<wsdls>
    <wsdl>
        <name>Partner 47.0 WSDL</name>
        <file>partner_47_0.wsdl</file>
    </wsdl>
    <wsdl>
        <name>Partner 46.0 WSDL</name>
        <file>partner_46_0.wsdl</file>
    </wsdl>
    <wsdl>
        <name>Partner 44.0 WSDL</name>
        <file>partner_44_0.wsdl</file>
    </wsdl>
    <wsdl>
        <name>Partner 43.0 WSDL</name>
        <file>partner_43_0.wsdl</file>
    </wsdl>
    <wsdl>
        <name>Partner 42.0 WSDL</name>
        <file>partner_42_0.wsdl</file>
    </wsdl>
    <wsdl>
        <name>Partner 41.0 WSDL</name>
        <file>partner_41_0.wsdl</file>
    </wsdl>
    <wsdl>
        <name>Partner 35.0 WSDL</name>
        <file>partner_35_0.wsdl</file>
    </wsdl>
    <wsdl>
        <name>Partner 34.0 WSDL</name>
        <file>partner_34_0.wsdl</file>
    </wsdl>
    <wsdl>
        <name>Partner 33.0 WSDL</name>
        <file>partner_33_0.wsdl</file>
    </wsdl>
```

SalesforceResources Folder

If the WSDL is successfully selected in an application module, the **SalesforceResources** folder is displayed in the Project Explorer view. It is used to store Salesforce resource files related to the current application module. The resource files include the WSDL files, metadata files, and so on.

You cannot rename, edit, or delete the WSDL and the metadata files in the **SalesforceResources** folder. But you can add other files to the **SalesforceResources** folder.

Switch a WSDL

The plug-in contains two types of Salesforce.com WSDLs, the Partner WSDL and the Enterprise WSDL. You can only select one WSDL for an application module. If you want to switch from the current WSDL to another, perform the Select WSDL operation again and follow these guidelines:

- When switching WSDLs between two different versions of Partner WSDLs, perform the Select WSDL operation to import a target WSDL first, and then update the metadata from the Salesforce.com server. For more information, see Refreshing Metadata.

- When the Partner WSDL is replaced with an Enterprise WSDL, all metadata files related to the original Partner WSDL are erased from the **SalesforceResources** folder. The Enterprise WSDL contains its own metadata that can be used in your process.

- When the Enterprise WSDL is replaced with a Partner WSDL, you must select a Partner WSDL first, and then perform the Getting Metadata operation to get the metadata from the Salesforce.com server.

- When the WSDL is changed, you are required to update the string in the **Server URL** field in the Salesforce Connection window by clicking the **Reload URL** button.

   - If the field is a Module property, you need to change the string in the Module Properties window manually. The Server URL in all the Salesforce configuration resources must be in the same version as the selected WSDL.

   - After the WSDL is changed, follow the steps in the popup after Select WSDL is complete. Open all Salesforce Connections that reference the older WSDL and click **Reload WSDL** and test and save the connections.

- For the Salesforce activities, the schema of the **Input**, **Output** and **Fault** tabs are parsed from the certain WSDL. After the WSDL is changed, the mapping configuration for the previously mentioned fields might be affected. In this case you have to repair any incorrect mappings.

Getting Metadata

Salesforce Metadata resource describes the schema definition metadata for a given organization as a XML Schema Definition (XSD).

Prerequisites

Before getting metadata for your project, you must create a Salesforce connection, with which you can download and store metadata as an XSD resource in the TIBCO Business Studio project during design time. For more information about creating a Salesforce connection, see Create a Salesforce Connection.

If you select an Enterprise WSDL, you need not perform the Get Metadata operation to get Salesforce metadata. The Enterprise WSDL contains Salesforce metadata. However, for bulk operations and composite API operations, Get Metadata prompts to get the Salesforce Bulk API and Salesforce Composite API metadata to use the bulk activities and composite activities.

Procedure

1. In the Project Explorer view, right-click the application module and select **BW Salesforce Tools** > **Get Metadata** .

2. Click **browse** next to the **Salesforce Connection** field to select one usable connection for the current activity.

3. Click **OK** in the Select Salesforce Connection window.

4. Click **OK** in the Get Salesforce Metadata window.

Result

The `Salesforce_Metadata.xsd`, `AsyncAPI.xsd` and `sforce_partner.xsd` schemas are displayed in the **SalesforceResources** folder of the application module in the Project Explorer view:

- The `Salesforce_Metadata.xsd` schema comes from a specified database on the Salesforce.com server.

- The Bulk API service is described by an XSD file. This XSD contains the schemas required for bulk operations. This file is downloaded and stored by default in `AsyncAPI.xsd` in the `SalesforceResources` folder when you fetch the metadata using the Get Metadata tool. You can change the name of the file in which the schema is fetched during the operation of getting the metadata.

- The `sforce_partner.xsd` schema stores some public data properties.

- The `composite.xsd` schema is used to fetch metadata for the Composite Batch activity.

- The `composite_dep.xsd` schema is used to fetch metadata for the Composite Dependent activity.

Refreshing Metadata

The Refreshing Metadata operation can be used to retrieve updated schemas from the Salesforce.com server.

Schema objects might change between the development phase and the production phase, or change over time. It is important to decide whether to accept or reject changes. If you accept a change, you are warned that all processes referring to the metadata might be impacted.

Procedure

1. Right-click the application module in the Project Explorer view, and then click **BW Salesforce Tools** > **Refresh Metadata**.

2. Click **OK** in the Refresh Salesforce Metadata window to perform the operation.

   - If no update of the metadata on the Salesforce.com server is available, then just click the relevant buttons to finish the operation.

   - If the metadata on the Salesforce.com server is different from the local metadata, a window opens, as shown in the following figure. Fields changed on the Salesforce.com server are displayed in the window.
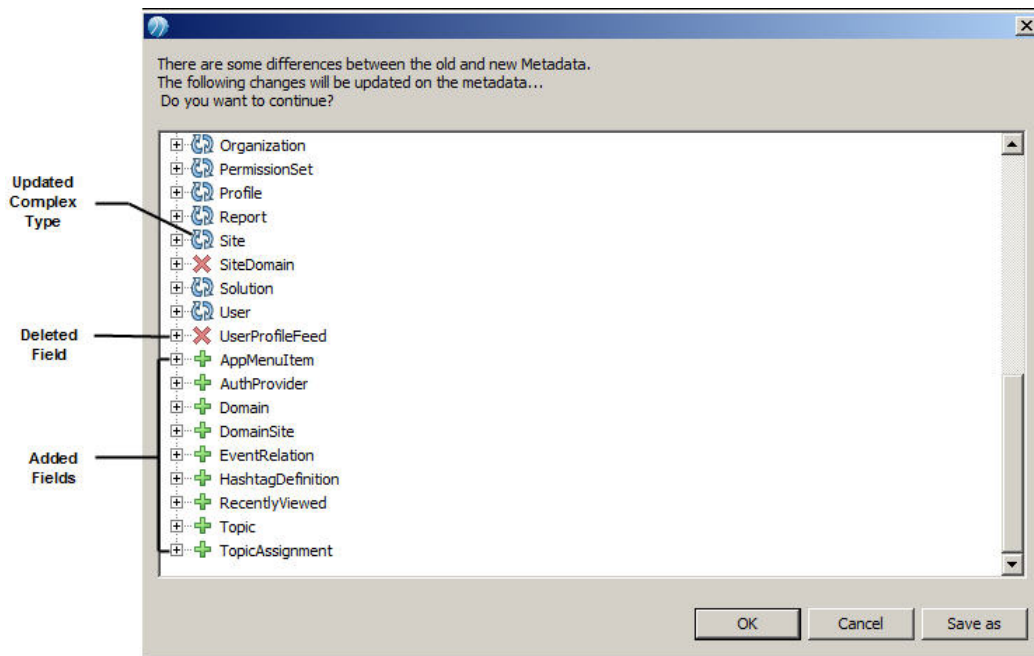
3. Click relevant buttons to perform corresponding operations.

    -   Click **OK** to accept the updated result.

    -   Click **Cancel** to reject this result.

    -   Click **Save As** to save the result to an `.xml` file.

Salesforce Connection

The Salesforce Connection shared resource specifies the Salesforce connection information.

General

The following table lists the fields on the **General** tab of the Salesforce Connection shared resource:

Field	Module Property?	Description
Package	No	The name of the package that contains the Salesforce Connection shared resource.
Name	No	The name displayed as the label for the shared resource.
Description	No	Add a short description for the shared resource.

Connection Configuration

The following table lists the fields on the **Connection Configuration** tab of the Salesforce Connection shared resource:

Field	Module Property?	Description
Server URL	Yes	The URL you use when connecting to the Salesforce.com login server. After selecting a WSDL for your project, the URL information is automatically filled in this field. If you reimport a different WSDL for your project, you must click **Reload WSDL** to refresh the WSDL URL for your project. You must maintain one Server URL when configuring the whole Salesforce project.
User Name	Yes	The user name to access the Salesforce.com server.
Password	Yes	The password to access the Salesforce.com server. When you connect through the plug-in, append the security token to your password. A security token is an automatically-generated key from Salesforce.com. For example, if your password is *mypassword*, and your security token is XXXXXXXXXX, then you must enter *mypassword*XXXXXXXXXX to log in. For detailed information on when to use a security token and how to generate it, see *salesforce.com Web Service API Developer's Guide*.
Session Timeout(Min)	Yes	Specifies the duration for which the session is activated. The value set here must be less than or equal to the value set on the Salesforce.com website. The default value is 120 minutes.

Field	Module Property?	Description
Test Connection	No	Click **Test Connection** to test whether the specified configuration fields result in a valid connection to a database on the Salesforce.com server.

Security

The following table lists the fields on the **Security** tab of the Salesforce Connection shared resource:

Field	Module Property?	Description
SSL Client Configuration	No	Select this check box to enable the SSL with SSL Client Configuration.
SSL Client	No	Select an SSL Client shared resource to establish a connection between the plug-in and the specified Salesforce server. This field is available when the **SSL Client Configuration** check box is selected. One way SSL - Ensure that the correct root certificates are present at the `<TIBCO_HOME>\tibcojre64\1.8.0\lib\security\cacerts trust store`. Mutual Authentication - Provide Identity Store Provider and details of the key. To configure the SSL Client, refer to the HTTP Client section in the Shared Resource topic of *TIBCO ActiveMatrix BusinessWorks Palette Reference Guide*.

Persistence

The following table lists the fields on the **Persistence** tab of the Salesforce Connection shared resource:

Field	Module Property?	Description
Custom Database Configuration	No	Select this check box to support persistence on a custom database instead of the engine database. For more information, see the description of **Persist ReplayID** field in the Salesforce Subscriber topic.
JDBC Resource	No	Select a JDBC shared resource with the custom database details configured.

Salesforce Palette

The Salesforce palette contains activities that can be added to your business processes. You can design a certain workflow with the Salesforce palette.

The Salesforce palette includes the following activities:

- Salesforce Create All
- Salesforce Delete All
- Salesforce Get Session
- Salesforce Query All
- Salesforce Retrieve All
- Salesforce Update All
- Salesforce Upsert All
- Salesforce Outbound Message Listener
- Salesforce Subscriber

Salesforce Create All

The Salesforce Create All activity adds one or more new individual objects to the database of your organization.

General

On the **General** tab, establish a connection with the Salesforce.com server.

The following table lists the configurations on the **General** tab of the Salesforce Create All activity:

Field	Module Property?	Description
Name	No	The name displayed as the label of the activity in the process.
Salesforce Connection	Yes	The path to the Salesforce shared resource. Click the **Choose/Create Default Resource** icon 🔍 to select a usable connection for the activity.

Description

On the **Description** tab, enter a short description for the Salesforce Create All activity.

Input

On the **Input** tab, you can specify input values for the Salesforce Create All activity.

The following table lists the input elements on the **Input** tab of the Salesforce Create All activity:

Input Items	Data Type	Description
connectionInfo (All fields in this section are optional.)		

Input Items	Data Type	Description
`serverUrl`	String	The web address of the endpoint that is used by this operation.
`sessionId`	String	The unique ID associated with this session.
`externalSession IdUsed`	Boolean	Specifies whether an external session ID is used (`true`) or not (`false`). If the value is set to `true`, an external session ID is filled in the **sessionId** field. In this case, when the session has to be refreshed, the refreshed operation cannot be performed, and an exception is thrown.
create		
`batchSize`	Integer	Optional. When processing large amounts of data, the activity internally invokes several SOAP calls. This field is used to set the batch size for the number of records created through a SOAP call. The value must be set between `1` and `200`. The default value is `200`. If the input value equals `-1`, it is set to the default value. If the input value is greater than `200`, it is set to `200`. If the input value is less than `1` (except `-1`), it is set to `1`.
`externalIdField Name`	String	Optional. This field is used when performing the retry operation. An internal upsert method is called to retry the operation. If this field is not specified when doing the retrying operation, an exception is thrown and the retrying operation is not performed. It contains the name of the field defined as a key field in the sObject.

`createSObjects`	Complex	Each `createSObjects` includes the following element:
		
		• `sObjects` (A complex value): Required. The complex sObject can be type substituted by a complex element from Salesforce_Metadata.xsd, retrieved from Salesforce.com server.
		
		    • In this release, you can create up to 10 object types in one call. Right-click **sObjects** > **Duplicate** to add multiple object types.
		    • You are not required to specify the **Id** field in the sObject.
		
		    To create a new relationship between a newly created sObject (sObject A) and an existing sObject (sObject B), where the sObject A schema includes the sObject B schema, set the value of the key field in sObject B while creating sObject A. If you modify the values of the fields in sObject B within the sObject A schema, the original sObject B is not modified. The key field in sObject B within the sObject A schema is only used to link sObject A with the original sObject B.

Input Items	Data Type	Description
create_Optional (All fields in this section are optional.)		
_configData		Each **_configData** includes the following element: • **timeout** (An integer value): the timeout value specifies the number of milliseconds an internal API call waits before the data returns. The default value is 15000 milliseconds. You can also change the default timeout value by setting the properties. See Default Timeout Value Setting for detailed information.

Input Items	Data Type	Description

headers		Each **headers** includes the following elements:
		
		 • **CallOptions**
		
		  − **client** (A string value): a string that identifies a particular client.
		
		  − **defaultNamespace** (A string value): a string that identifies a developer namespace prefix. Use this field to resolve field names in managed packages without having to fully specify the fieldName everywhere.
		
		 • **AssignmentRuleHeader**
		
		  − **assignmentRuleId** (A string value): the ID of a specific assignment rule to run for the case or lead. It can be an inactive assignment rule. The ID can be retrieved by querying an AssignmentRule object. If **assignmentRuleId** is specified, do not specify **useDefaultRule**. This element is ignored for accounts, because all territory assignment rules are applied.
		
		  The case and the lead are two predefined Salesforce.com schema types. For more information about these two types, see *Force.com Web Service API Developer's Guide*.
		
		  − **useDefaultRule** (A Boolean value): if the value is `true` for a case or a lead, the default (active) assignment rule is used for the case or the lead. If **useDefaultRule** is specified, do not specify **assignmentRuleId** and type an empty string in the **assignmentRuleId** field. If `true` for an account, all territory assignment rules are applied, and if `false`, no territory assignment rules are applied.
		
		 • **MruHeader**
		
		  − **updateMru** (A Boolean value): specifies whether to update the list of most recently used items (`true`) or not (`false`).
		
		 • **AllowFieldTruncationHeader**
		
		  − **allowFieldTruncation** (A Boolean value): specifies whether to truncate field values that are too long (`true`) or not (`false`).
		
		  Default is `false`: no change in behavior. If a string or textarea value is too large, the operation fails and the fault code `STRING_TOO_LONG` is returned.
		
		 • **DisableFeedTrackingHeader**
		
		  − **disableFeedTracking** (A Boolean value): if it is set to `true`, the changes made in the current call are not tracked in feeds.
		
		  The default is `false`.
		
		 • **StreamingEnabledHeader**

Input Items	Data Type	Description

- **streamingEnabled** (A Boolean value): specifies whether you want to receive streaming notifications for changes to Salesforce data.

- **AllOrNoneHeader**

  - **allOrNone** (A Boolean value): if **allOrNone** is set to `true`, any failed records in a call cause all changes for the call to be rolled back. Record changes are not committed unless all records are processed successfully.

    The default is `false`. Some records can be processed successfully while others are marked as failed in the call results.

    > When processing large amounts of data, the activity internally invokes several SOAP calls. However, this **AllOrNoneHeader** applies to each internal SOAP call individually. If **allOrNone** is set to `true`, only individual internal SOAP calls with failed records are rolled back. All the record changes of other calls are committed.

- **DuplicateRuleHeader**

  - **allowSave** (A Boolean value): if **allowSave** is set to `true`, duplicate records are saved. If this value is set to `false` duplicate records are prevented from being saved.

  - **includeRecordDetails** (A Boolean value): if **includeRecordDetails** is set to `true`, fields and values for records detected as duplicates are retrieved. If this value is set to `false`, only record IDs for records detected as duplicates are retrieved.

  - **runAsCurrentUser** (A Boolean value): if **runAsCurrentUser** is set to `true`, sharing rules for the current users are enforced when duplicate rules run. If this value is set to `false`, sharing rules specified in the class for the request are used.

    > If no sharing rules are specified, Apex code runs in system context and sharing rules for the current user are not enforced.

- **LocaleOptions**

  - **language** (A string value): specifies the language of the labels returned. The value must be a valid user locale (language and country), such as `de_DE` or `en_GB`. For more information about the locales, see *Force.com Web Service API Developer's Guide*.

- **DebuggingHeader**

  - **debugLevel** (A string value): specifies the level of detail in the debug header.

    See Salesforce.com document *Apex Developer's Guide* for detailed information.

Input Items	Data Type	Description
		The response of the debugging information can be found in the SOAP message log. You might have to enable the debug role log to trace the debugging errors. • **PackageVersionHeader** Each **PackageVersion** includes the following elements: • **majorNumber** (An integer value): the major version number of a package version. A package version is denoted by *majorNumber.minorNumber*, for example 2.1. • **minorNumber** (An integer value): the minor version number of a package version. A package version is denoted by *majorNumber.minorNumber*, for example 2.1. • **namespace** (A string value): the unique namespace of the managed package. • **EmailHeader** – **triggerAutoResponseEmail** (A Boolean value): specifies whether to trigger auto-response rules (true) or not (false), for leads and cases. In the Salesforce.com user interface, this email can be automatically triggered by a number of events, for example creating a case or resetting a user password. – **triggerOtherEmail** (A Boolean value): specifies whether to trigger email outside the organization (true) or not (false). In the Salesforce.com user interface, this email can be automatically triggered by creating, editing, or deleting a contact of a case. – **triggerUserEmail** (A Boolean value): specifies whether to trigger an email that is sent to users in the organization (true) or not (false). In the Salesforce user interface, this email can be automatically triggered by a number of events: resetting a password, creating a new user, adding comments to a case, creating a task, or modifying a task.

Output

On the **Output** tab, you can find the output value.

The following table lists the output elements on the **Output** tab of the Salesforce Create All activity:

Output Item		Data Type	Description
createAllResponse			
result	**errors**	Complex	If errors occur during the activity, an array of error objects with an error code and description is returned.

Output Item	Data Type	Description
`id`	String	The ID of an sObject that you attempt to create.
`success`	Boolean	Indicates whether the Create activity has succeeded (`true`) or not (`false`).

Fault

On the **Fault** tab, you can find the error code and error message of the Salesforce Create All activity. See Error Codes for more information about error codes and corrective actions to take.

Fault	Thrown when..
SalesforceExecuteSOAPMethodException	An error occurred when calling a SOAP method. For example, wrong values are set in the **Input** fields or the Salesforce session has timed out.
SalesforceLoginException	An error occurred when logging in to the Salesforce.com server.
SalesforceConnectionNotFoundException	An error occurred when the external session ID was not used or the Salesforce connection was not configured correctly.

Salesforce Delete All

The Salesforce Delete All activity deletes one or more individual objects from the database of your organization.

General

On the **General** tab, you can establish a connection to the Salesforce.com server.

The following table lists the configurations on the **General** tab of the Salesforce Delete All activity:

Field	Module Property?	Description
Name	No	The name displayed as the label of the activity in the process.
Salesforce Connection	Yes	The path to the Salesforce shared resource. Click the **Choose/Create Default Resource** icon [Q] to select a usable connection for the activity.

Description

On the **Description** tab, you can enter a short description for the Salesforce Delete All activity.

Input

On the **Input** tab, you can specify input values for the Salesforce Delete All activity.

The following table lists the input elements on the **Input** tab of the Salesforce Delete All activity:

Input Items	Data Type	Description
connectionInfo (All fields in this section are optional.)		
serverUrl	String	The web address of the endpoint that is used by this operation.
sessionId	String	The unique ID associated with this session.
externalSessionIdUsed	Boolean	Specifies whether an external session ID is used (`true`) or not (`false`). If the value is set to `true`, an external session ID is filled in the **sessionId** field. In this case, when the session has to be refreshed, the refreshed operation cannot be performed, and an exception is thrown.
delete		Each **delete** includes the following elements: • **batchSize** (An integer value): optional. When processing large amounts of data, the activity internally invokes several SOAP calls. This field is used to set the batch size for the number of records deleted through an SOAP call. The value must be set between `1` and `200`. The default value is `200`. If the input value equals `-1`, it is set to the default value. If the input value is greater than `200`, it is set to `200`. If the input value is less than `1` (except `-1`), it is set to `1`. • **deleteSObjects** – **ids** (A string value): required. An array of one or more IDs associated with the objects to be deleted.
delete_Optional (All fields in this section are optional.)		
_configData		Each **_configData** includes the following node: • **timeout** (An integer value): the **timeout** value specifies the number of milliseconds an internal API call waits before the data returns. The default value is `15000` milliseconds. You can also change the default timeout value by setting the properties. See Default Timeout Value Setting for detailed information.

Input Items	Data Type	Description
`headers`		Each `headers` includes the following elements:

Each `headers` includes the following elements:

- **`CallOptions`**

  - **`client`** (A string value): a string that identifies a particular client.

  - **`defaultNamespace`** (A string value): a string that identifies a developer namespace prefix. Use this field to resolve field names in managed packages without having to fully specify the fieldName everywhere.

- **`PackageVersionHeader`**

  Each **`PackageVersion`** includes the following nodes:

  - **`majorNumber`** (An integer value): the major version number of a package version.

    A package version is denoted by *majorNumber.minorNumber*, for example 2.1.

  - **`minorNumber`** (An integer value): the minor version number of a package version.

    A package version is denoted by *majorNumber.minorNumber*, for example 2.1.

  - **`namespace`** (A string value): the unique namespace of the managed package.

- **`UserTerritoryDeleteHeader`**

  - **`transferToUserId`** (A string value): the ID of the user to whom open opportunities in that user's territory is assigned when an opportunity's owner (user) is removed from a territory.

- **`EmailHeader`**

  - **`triggerAutoResponseEmail`** (A Boolean value): specifies whether to trigger auto-response rules (`true`) or not (`false`), for leads and cases. In the Salesforce.com user interface, this email can be automatically triggered by a number of events, for example creating a case or resetting a user password.

  - **`triggerOtherEmail`** (A Boolean value): specifies whether to trigger email outside the organization (`true`) or not (`false`). In the Salesforce.com user interface, this email can be automatically triggered by creating, editing, or deleting a contact of a case.

  - **`triggerUserEmail`** (A Boolean value): specifies whether to trigger email that is sent to users in the organization (`true`) or not (`false`). In the Salesforce user interface, this email can be automatically triggered by a number of events: resetting a password, creating a new user, adding comments to a case, or creating or modifying a task.

- **`AllowFieldTruncationHeader`**

28

Input Items	Data Type	Description

    – **allowFieldTruncation** (A Boolean value): specifies whether to truncate field values that are too long (`true`) or not (`false`).

    Default is `false`: no change in behavior. If a string or textarea value is too large, the operation fails and the fault code `STRING_TOO_LONG` is returned.

- **DisableFeedTrackingHeader**

    – **disableFeedTracking** (A Boolean value): if it is set to `true`, the changes made in the current call are not tracked in feeds.

    The default is `false`.

- **StreamingEnabledHeader**

    – **streamingEnabled** (A Boolean value): specifies whether you want to receive streaming notifications for changes to Salesforce data.

- **AllOrNoneHeader**

    – **allOrNone** (A Boolean value): if **allOrNone** is set to `true`, any failed records in a call cause all changes for the call to be rolled back. Record changes are not committed unless all records are processed successfully.

    The default is `false`. Some records can be processed successfully while others are marked as failed in the call results.

> When processing large amounts of data, the activity internally invokes several SOAP calls.
>
> However, this **AllOrNoneHeader** applies to each internal SOAP call individually. If **allOrNone** is set to `true`, only individual internal SOAP calls with failed records are rolled back. All the record changes of other calls are committed.

- **DuplicateRuleHeader**

    – **allowSave** (A Boolean value): if **allowSave** is set to `true`, duplicate records are saved. If this value is set to `false` duplicate records are prevented from being saved.

    – **includeRecordDetails** (A Boolean value): if **includeRecordDetails** is set to `true`, fields and values for records detected as duplicates are retrieved. If this value is set to `false`, only record IDs for records detected as duplicates are retrieved.

    – **runAsCurrentUser** (A Boolean value): if **runAsCurrentUser** is set to `true`, sharing rules for the current user are enforced when duplicate rules run. If this value is set to `false`, sharing rules specified in the class for the request are used.

> If no sharing rules are specified, Apex code runs in system context and sharing rules for the current user are not enforced.

Input Items	Data Type	Description
		• **LocaleOptions**
		– **language** (A string value): specifies the language of the labels returned. The value must be a valid user locale (language and country), such as de_DE or en_GB. For more information about the locales, see *Force.com Web Service API Developer's Guide*.
		• **DebuggingHeader**
		– **debugLevel** (A string value): specifies the level of detail in the debug header. See Salesforce.com document *Apex Developer's Guide* for detailed information. The response of debugging information can be found in the SOAP message log. You might have to enable the debug role log to trace debugging errors.

Output

On the **Output** tab, you can find the output value.

The following table lists the output elements on the **Output** tab of the Salesforce Delete All activity:

Output Item		Data Type	Description
deleteAllResponse			
result	**errors**	Complex	If errors occur during the activity, an array of error objects (with an error code and description) are returned.
	id	String	The ID of an sObject that you attempted to delete.
	success	Boolean	Indicates whether the delete activity has succeeded (true) or not (false).

Fault

On the **Fault** tab, you can find the error code and error message of the Salesforce Delete All activity. See Error Codes for more information about error codes and corrective actions to take.

Fault	Thrown when
SalesforceExecuteSOAPMethodException	An error occurs when calling a SOAP method. For example, wrong values are set in the **Input** fields or the Salesforce session times out.
SalesforceLoginException	An error occurs when logging in to the Salesforce.com server.
SalesforceConnectionNotFoundException	An error occurs when the external session ID is not used or the Salesforce connection is not configured correctly.

Salesforce Get Session

The Salesforce Get Session activity retrieves an object reference for a specified Salesforce connection and gets an existing session. This session can be used within Salesforce activities to access a specified database.

General

On the **General** tab, you can establish a connection to the Salesforce.com server.

The following table lists the configurations on the **General** tab of the Salesforce Get Session activity:

Field	Module Property ?	Description
Name	No	The name displayed as the label of the activity in the process.
Salesforce Connection	Yes	The path to the Salesforce shared resource. Click the **Choose/Create Default Resource** icon 🔍 to select a usable connection for the activity.

Description

On the **Description** tab, you can enter a short description for the Salesforce Get Session activity.

Input

On the **Input** tab, you can specify input values for the Salesforce Get Session activity.

The following table lists the input elements on the **Input** tab of the Salesforce Get Session activity:

Input Item	Data Type	Description
login		
`salesforceConnection`	String	Optional. The path to the shared resource which contains the Salesforce connection information. The priority of this field is higher than the **Salesforce Connection** field on the **General** tab. If you want to change the referenced Salesforce Connection shared resource at run time, you can set it here.
`refreshSession`	Boolean	Optional. Specify whether a new session can be initiated. Set the value to `true` if you always want to get a new session. If the value is set to `false`, the new session is not used unless the existing session has timed out. The performance time in this situation is shorter.

Output

On the **Output** tab, you can find the output value.

The following table lists the output elements on the **Output** tab of the Salesforce Get Session activity:

Output Item	Data Type	Description
`salesforceConnection`	String	The path to the shared resource which contains the Salesforce connection information. The Salesforce Get Session activity uses this Salesforce Connection shared resource at run time.
`loginResponse`		
`result`		Each `result` includes the following elements: • `metadataServerUrl` (A string value): the web address of the endpoint which processes subsequent metadata API calls. • `passwordExpired` (A Boolean value): indicates whether the password used during the login attempt has expired (`true`) or not (`false`). • `sandbox` (A Boolean value): specifies whether the using Salesforce environment is a `sandbox` or not. • `serverUrl` (A string value): the web address of the endpoint which processes subsequent API calls. • `sessionId` (A string value): the unique ID associated with this session. • `userId` (A string value): the ID of the user associated with the specified user name and password. • `userInfo` (A complex value): information fields related to the user.

Fault

On the **Fault** tab, you can find the error code and error message of the Salesforce Get Session activity. See Error Codes for more information about error codes and corrective actions to take.

Fault	Thrown when
SalesforceLoginException	An error occurred when logging in to the Salesforce.com server.
SalesforceConnectionNotFoundException	An error occurred when the external session ID is not used or the Salesforce connection is not configured correctly.

Salesforce Query All

The Salesforce Query All activity performs the specified SOQL SELECT statement. It executes a query against the specified object and returns the data that matches the specified criteria.

General

On the **General** tab, you can establish a connection to the Salesforce.com server.

The following table lists the configurations on the **General** tab of the Salesforce Query All activity:

Field	Module Property?	Description
Name	No	The name displayed as the label of the activity in the process.
Salesforce Connection	Yes	The path to the Salesforce shared resource. Click the **Choose/Create Default Resource** icon 🔍 to select a usable connection for the activity.
Query Type	Yes	Select a query type from the drop-down list. The default type is Query. If you select **Query** from the drop-down list, the operation retrieves existing records only. If you select **QueryAll**, the operation retrieves all archived and deleted records. It can be provided using both module and process property.
Salesforce Query	No	Displays the query to be executed against the specified object. Click **Query Builder** to open the wizard to build your query or clear the existing query with **Clear Query**. For more information about the query builder, see Salesforce Query Builder.
Prepared Statement	No	If you use a prepared parameter '?' in the query, then define the parameter name and its data type in the statement. You can add multiple prepared parameters using the add button.

Description

On the **Description** tab, you can enter a short description for the Salesforce Query All activity.

Input

On the **Input** tab, you can specify input values for the Salesforce Query All activity.

The following table lists the input elements on the **Input** tab of the Salesforce Query All activity:

Input Items	Data Type	Description
`connectionInfo` (All fields in this section are optional.)		
`serverUrl`	String	The web address of the endpoint that is used by this operation.
`sessionId`	String	The unique ID associated with this session.
`externalSessionIdUsed`	Boolean	Specifies whether an external session ID is used (`true`) or not (`false`). If the value is set to `true`, an external session ID is filled in the **sessionId** field. In this case, when the session has to be refreshed, the refreshed operation cannot be performed, and an exception is thrown.
`query` (All fields in this section are optional.)		

Input Items	Data Type	Description
batchSize	Integer	When processing large amounts of data, the activity internally invokes several SOAP calls. This field is used to set the batch size for the number of records returned through an SOAP call.
		The value must be set between 200 and 2000. The default value is 500.
		If the input value equals -1, it is set to the default value. If the input value is greater than 2000, it is set to 2000. If the input value is less than 200 (except -1), it is set to 200.
queryMain		Each **queryMain** includes the following element:
		• **queryString** (A string value): required. The query string follows SOQL syntax. It specifies the object to query, the fields to return, and any condition of including a specific object in the query.
fields		This field is visible when you define a prepared parameter on the **General** tab. You can pass the prepared parameters values under the parameters.
query_Optional (All fields in this section are optional.)		
_configData		Each **_configData** includes the following element:
		• **timeout** (An integer value): the **timeout** value specifies the number of milliseconds an internal API call waits before the data returns.
		The default value is 15000 milliseconds.
		You can also change the default timeout value by setting the properties. See Default Timeout Value Setting for detailed information.

Input Items	Data Type	Description

`headers`		Each `headers` includes the following elements:
		
		• `CallOptions`
		
		  – `client` (A string value): a string that identifies a particular client.
		
		  – `defaultNamespace` (A string value): a string that identifies a developer namespace prefix. Use this field to resolve field names in managed packages without having to fully specify the fieldName everywhere.
		
		• `MruHeader`
		
		  – `updateMru` (A Boolean value): specifies whether to update the list of most recently used items (`true`) or not (`false`), which is on the sidebar of the Salesforce.com user interface.
		
		• `PackageVersionHeader`
		
		Each `PackageVersion` includes the following elements:
		
		• `majorNumber` (An integer value): the major version number of a package version.
		
		A package version is denoted by *majorNumber.minorNumber*, for example 2.1.
		
		• `minorNumber` (An integer value): the minor version number of a package version.
		
		A package version is denoted by `majorNumber.minorNumber`, for example 2.1.
		
		• `namespace` (A string value): the unique namespace of the managed package.

Input Items	Data Type	Description

`SubsetSetting`		Each `SubsetSetting` includes the following elements:
		
		• `ProcessInSubsets`(A Boolean value): optional. Specifies whether you can process smaller batches of rows instead of retrieving one large result set (`true`) or not (`false`).
		
		  This element must be used in conjunction with a Repeat loop scope to process the result sets. See Querying Data in Subset Mode for more information.
		
		  The default value is `false`.
		
		• `subsetSize` (An integer value): optional. This element is only available when the **ProcessInSubsets** field is set to `true`. When that field is set to `true`, records are processed in batches. The returned data is passed out from the activity when completing the activity in a `subsetSize` sized batch.
		
		  This element specifies the maximum number of messages that are picked up for each execution loop. Its region is from `1` to `10`.
		
		  You can process smaller batches of rows instead of retrieving one large result set by using this element. This element must be used in conjunction with a Repeat loop scope to process the result sets. See Querying Data in Subset Mode for more information.
		
		  The default value is `1`.

Output

On the **Output** tab, you can find the output value.

The following table lists the output elements on the **Output** tab of the Salesforce Query All activity:

Output Item		Data Type	Description
`queryAllResponse`			
`result`	`done`	Boolean	Indicates whether additional rows are to be retrieved from the query results (`false`) using another query activity, or not (`true`). Your client application can use this value as a loop condition when iterating through the query results.
	`queryLocator`	String	Used in subsequent query activities for retrieving sets of objects from the query results, if applicable.
	`records`	Complex	An array of sObjects representing individual objects of the specified object and containing data defined in the `fieldlist` specified in the `queryString` element. These sObjects can be converted to the specified type defined in the metadata schema, for example, the Salesforce_Metadata, retrieved from the Salesforce.com server.

Output Item	Data Type	Description
`size`	Integer	Total number of rows retrieved in the query. Your client application can use this value to determine whether the query retrieved any rows (size > 0) or not (size = 0).
`lastSubset`	Boolean	Indicates whether the current output is the last subset (`true`) or not (`false`).

Fault

On the **Fault** tab, you can find the error code and error message of the Salesforce Query All activity. See Error Codes for more information about error codes and corrective actions to take.

Fault	Thrown when
SalesforceExecuteSOAPMethodException	An error occurred when calling an SOAP method. For example, wrong values are set in the **Input** fields or the Salesforce session has timed out.
SalesforceLoginException	An error occurred when logging in to the Salesforce.com server.
SalesforceConnectionNotFoundException	An error occurred when the external session ID is not used or the Salesforce connection is not configured correctly.

🛈 Projects created in Salesforce Plug-in version 6.1.1 or earlier have an input/output validation issue for Salesforce Query All activity, due to the namespace change in Salesforce Plug-in version 6.2.0. This can be fixed using the mapper check and repair.

Querying Data in Subset Mode

When the result set of a query is very large, you can query subsets of the result set and iterate until the entire result set is processed.

To query subsets, you must use a Repeat loop scope to iterate through the entire result set:

Procedure

1. Create a Salesforce Query All activity.

2. Specify the fields on the **General** tab and create a query that returns multiple rows.

3. Set the **ProcessInSubsets** element of the **Input** tab to `true`.

4. Set the **subsetSize** element of the **Input** tab to the maximum number of messages you want to process for each execution loop.

5. Right-click the **SalesforceQueryAll** activity icon, and then click **Create Group** > **Scope** in the pop-up menu to create a scope containing the activity.
   A cyclic line and a curve with an arrow are displayed around the **SalesforceQueryAll** activity icon.

6. Click the cyclic line to select it.

7. Select **Repeat** from the **Group Type** list on the **General** tab, and specify an index name.

8. A loop exits when the entire result set is consumed. You can set the condition of the loop by editing the **lastSubset** element on the **Output** tab.

> The previous procedure is a general guideline for creating a loop scope to process a large set of records. You might want to modify the procedure to include additional processing of the records, or you might want to change the XPath expressions to suit your business process.

Relationship Query

The Relationship Query can be used to query multiple sObjects concurrently in the Salesforce.com database.

A relationship is created between those sObjects. Relationship queries follow the SOQL syntax. Parent-to-children and child-to-parent are the two possible relationships between sObjects. In the Salesforce.com database, some sObjects with a relationship are predefined. For example, Account is a parent to Contact. You can use relationship queries to search for sObjects of one type based on criteria that applies to sObjects of another type. For example, "return all accounts created by Bob Jones and the contacts associated with those accounts." The sObjects are connected either by parent-to-children or child-to-parent relationship, as shown in the following figure:
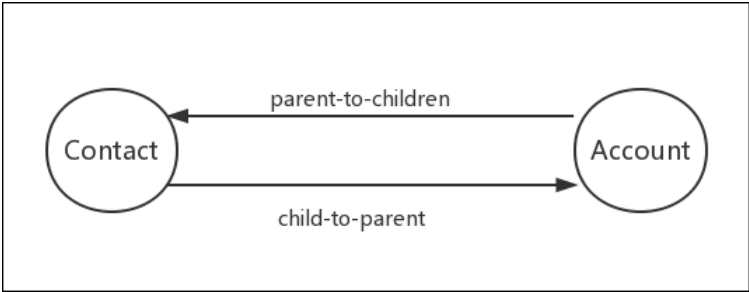
For more information about relationship queries, See *Force.com Web Service API Developer's Guide* and Working With RelationshipQuery Project.

Salesforce Retrieve All

The Salesforce Retrieve All activity retrieves one or more individual objects based on the specified object IDs from the database of your organization.

General

On the **General** tab, you can establish a connection to the Salesforce.com server.

The following table lists the configurations on the **General** tab of the Salesforce Retrieve All activity:

Field	Module Property?	Description
Name	No	The name displayed as the label of the activity in the process.
Salesforce Connection	Yes	The path to the Salesforce shared resource. Click the **Choose/Create Default Resource** icon 🔍 to select a usable connection for the activity.

Description

On the **Description** tab, you can enter a short description for the Salesforce Retrieve All activity.

Input

On the **Input** tab, you can specify input values for the Salesforce Retrieve All activity.

The following table lists the input elements on the **Input** tab of the Salesforce Retrieve All activity:

Input Items	Data Type	Description
connectionInfo (All fields in this section are optional.)		
serverUrl	String	The web address of the endpoint that is used by this operation.
sessionId	String	The unique ID associated with this session.
externalSession IdUsed	Boolean	Specifies whether an external session ID is used (`true`) or not (`false`).
		If the value set to `true`, an external session ID is filled in the **sessionId** field.
		In this case, when the session has to be refreshed, the refreshed operation cannot be performed, and an exception is thrown.
retrieve		
batchSize	Integer	Optional. When processing large amounts of data, the activity internally invokes several SOAP calls. This field is used to set the batch size for the number of records returned through an SOAP call.
		The value must be set between `200` and `2000`. The default value is `500`.
		If the input value equals `-1`, it is set to the default value. If the input value is greater than `2000`, it is set to `2000`. If the input value is less than `200` (except `-1`), it is set to `200`.
retrieveMain		Each **retrieveMain** includes the following elements: **fieldList** (A string value): required. A list of one or more fields in the specified object, separated by commas. You must specify valid field names and must have read-level permissions to each specified field. The **fieldList** defines the ordering of fields in the result. The usage of the wildcard (*) might cause a very large result set to be returned, which can slow the client application performance, so it is best practice to use this function when necessary. **sObjectType** (A string value): required. The specified value must be a valid object for your organization. **ids** (A string value): required. An array of one or more IDs of the objects to retrieve. You can pass a maximum of 2000 object IDs to the activity.

Input Items	Data Type	Description
retrieve_Optional		
_configData		Each _configData includes the following element: • timeout (An integer value): the timeout value specifies the number of milliseconds an internal API call waits before the data returns. The default value is 15000 milliseconds. You can also change the default timeout value by setting the properties. See Default Timeout Value Setting for detailed information.
headers		Each headers includes the following elements: • CallOptions – client (A string value): a string that identifies a particular client. – defaultNamespace (A string value): a string that identifies a developer namespace prefix. Use this field to resolve field names in managed packages without having to fully specify the fieldName everywhere. • MruHeader – updateMru (A Boolean value): specifies whether to update the list of most recently used items (true) or not (false), which is on the sidebar of the Salesforce.com user interface. • PackageVersionHeader Each PackageVersion includes the following nodes: • majorNumber (An integer value): the major version number of a package version. A package version is denoted by *majorNumber.minorNumber*, for example 2.1. • minorNumber (An integer value): the minor version number of a package version. A package version is denoted by *majorNumber.minorNumber*, for example 2.1. • namespace (A string value): the unique namespace of the managed package.

Input Items	Data Type	Description

SubsetSetting		Each **SubsetSetting** includes the following elements:
		
		• **ProcessInSubsets** (A Boolean value): optional. Specifies whether you can process smaller batches of rows instead of retrieving one large result set (`true`) or not (`false`).
		
		  This element must be used in conjunction with a Repeat loop scope to process the result sets. See Retrieving Data in the Subset Mode for more information.
		
		  The default value is `false`.
		
		• **subsetSize** (An integer value): optional. This element is only available when the **ProcessInSubsets** field is set to `true`. When that field is set to `true`, records are processed in batches. The returned data is passed out from the activity when completing the activity in a **subsetSize** sized batch.
		
		  This element specifies the maximum number of messages that are picked up for each execution loop. Its region is from `1` to `10`.
		
		  You can process smaller batches of rows instead of retrieving one large result set by using this element. This element must be used in conjunction with a Repeat loop scope to process the result sets. See Retrieving Data in the Subset Mode for more information.
		
		  The default value is `1`.

Output

On the **Output** tab, you can find the output value.

The following table lists the output elements on the **Output** tab of the Salesforce Retrieve All activity:

Output Item	Data Type	Description
retrieveAllResponse		
result	Complex	An array of sObjects representing individual objects of the specified object and containing data defined in the **fieldlist** specified in the **ids**. Those sObjects can be converted to the specified type defined in the metadata schema, for example, the Salesforce_Metadata, retrieved from the Salesforce.com server.
lastSubset	Boolean	Indicates whether the current output is the last subset (`true`) or not (`false`).

Fault

On the **Fault** tab, you can find the error code and error message of the Salesforce Retrieve All activity. See Error Codes for more information about error codes and corrective actions to take.

Fault	Thrown when
SalesforceExecuteSOAPMethodException	An error occurred when calling an SOAP method. For example, wrong values are set in the Input fields or the Salesforce session has timed out.
SalesforceLoginException	An error occurred when logging in to the Salesforce.com server.
SalesforceConnectionNotFoundException	An error occurred when the external session ID is not used or the Salesforce connection is not configured correctly.

Retrieving Data in the Subset Mode

When the result set of a retrieve is very large, you can retrieve subsets of the result set and iterate until the entire result set is processed.

To retrieve subsets, you must use a Repeat loop scope to iterate through the entire result set:

Procedure

1. Create a Salesforce Retrieve All activity.

2. Specify the fields on the **General** tab and create a retrieve that returns multiple rows.

3. Set the **ProcessInSubsets** element of the **Input** tab to `true` .

4. Set the **subsetSize** element of the **Input** tab to the maximum number of messages you want to process for each execution loop.

5. Right-click the **SalesforceQueryAll** activity icon, and then click **Create Group** > **Scope** from the pop-up menu to create a scope containing the activity.
   A cyclic line and a curve with an arrow appear around the **SalesforceQueryAll** activity icon.

6. Click the cyclic line to select it.

7. Select **Repeat** from the **Scope Type** list on the **General** tab, and specify an index name.

8. The loop exits when the entire result set has been consumed. You can set the condition of the loop by editing the `lastSubset` element on the **Output** tab.

> The previous procedure is a general guideline for creating a loop scope to process a large set of records. You might want to modify the procedure to include additional processing of the records, or you might want to change the XPath expressions to suit your business process.

Salesforce Update All

The Salesforce Update All activity updates one or more existing objects within the database of your organization.

General

On the **General** tab, you can establish a connection to the Salesforce.com server.

The following table lists the configurations on the **General** tab of the Salesforce Update All activity:

Field	Module Property?	Description
Name	No	The name displayed as the label of the activity in the process.
Salesforce Connection	Yes	The path to the shared configuration resource defining how the application joins the Salesforce. Click the **Choose/Create Default Resource** icon 🔍 to select a usable connection for the activity.

Description

On the **Description** tab, you can enter a short description for the Salesforce Update All activity.

Input

On the **Input** tab, you can specify input values for the Salesforce Update All activity.

The following table lists the input elements on the **Input** tab of the Salesforce Update All activity:

Input Items	Data Type	Description
connectionInfo (All fields in this section are optional.)		
`serverUrl`	String	The web address of the endpoint that is used by this operation.
`sessionId`	String	The unique ID associated with this session.
`externalSessionIdUsed`	Boolean	Specifies whether an external session ID is used (`true`) or not (`false`). If the value is set to `true`, an external session ID is filled in the **sessionId** field. In this case, when the session has to be refreshed, the refreshed operation cannot be performed, and an exception is thrown.
`update`		
`batchSize`	Integer	Optional. When processing large amounts of data, the activity internally invokes several SOAP calls. This field is used to set the batch size for the number of records updated through an SOAP call. The value must be set between `1` and `200`. The default value is `200`. If the input value equals `-1`, it is set to the default value. If the input value is greater than `200`, it is set to `200`. If the input value is less than `1` (except `-1`), it is set to `1`.

Input Items	Data Type	Description
updateSObjects		Each **updateSObjects** includes the following element: • **sObjects** (A complex value): required. An array of one or more objects to update. This sObject can be converted to the specified type defined in the metadata schema, for example, the Salesforce_Metadata, retrieved from the Salesforce.com server. To create a new relationship between a newly created sObject (sObject A) and an existing sObject (sObject B), where the sObject A schema includes the sObject B schema, set the value of the key field in sObject B while updating sObject A. If you modify the values of the fields in sObject B within the sObject A schema, the original sObject B is not modified. The key field in sObject B within the sObject A schema is only used to link sObject A with the original sObject B. 📝 In this release, you can create up to 10 object types in one call. Click **sObjects** > **Duplicate** to add multiple object types.
update_Optional		
_configData		Each **_configData** includes the following element: • **timeout** (An integer value): the timeout value specifies the number of milliseconds an internal API call waits before the data returns. The default value is 15000 milliseconds. You can also change the default timeout value by setting the properties. See Default Timeout Value Setting for detailed information.

Input Items	Data Type	Description
headers		Each **headers** includes the following elements:

Each **headers** includes the following elements:

- **CallOptions**

  - **client** (A string value): a string that identifies a particular client.

  - **defaultNamespace** (A string value): a string that identifies a developer namespace prefix. Use this field to resolve field names in managed packages without having to fully specify the fieldName everywhere.

- **AssignmentRuleHeader**

  - **assignmentRuleId** (A string value): the ID of a specific assignment rule to run for the case or the lead. It can be an inactive assignment rule. The ID can be retrieved by querying an AssignmentRule object. If specified, do not specify useDefaultRule. This element is ignored for Accounts, because all territory assignment rules are applied.

    The case and the lead are two pre-defined Salesforce.com schema types. For more information about these two types, see *Force.com Web Service API Developer's Guide*.

  - **useDefaultRule** (A Boolean value): if `true` for a case or a lead, uses the default (active) assignment rule for the case or the lead. If specified, do not specify an **assignmentRuleId** and type an empty string in the **assignmentRuleId** field. If `true` for an Account, all territory assignment rules are applied, and if `false`, no territory assignment rules are applied.

- **MruHeader**

  - **updateMru** (A Boolean value): specifies whether to update the list of most recently used items (`true`) or not (`false`), which is on the sidebar of the Salesforce.com user interface.

- **AllowFieldTruncationHeader**

  - **allowFieldTruncation** (A Boolean value): specifies whether to truncate field values that are too long (`true`) or not (`false`).

    Default value is `false`: no change in behavior. If a string or text area value is too large, the operation fails and the fault code `STRING_TOO_LONG` is returned.

- **DisableFeedTrackingHeader**

  - **disableFeedTracking** (A Boolean value): if it is set to `true`, the changes made in the current call are not tracked in feeds.

    The default is `false`.

- **StreamingEnabledHeader**

Input Items	Data Type	Description
		— **streamingEnabled** (A Boolean value): specifies whether you want to receive streaming notifications for changes to Salesforce data.

- **AllOrNoneHeader**

  – **allOrNone** (A Boolean value): if **allOrNone** is set to true, any failed records in a call cause all changes for the call to be rolled back. Record changes are not committed unless all records are processed successfully.

    The default value is false. Some records can be processed successfully while others are marked as failed in the call results.

    > When processing large amounts of data, the activity internally invokes several SOAP calls.
    >
    > However, this **AllOrNoneHeader** applies to each internal SOAP call individually. If **allOrNone** is set to true, only individual internal SOAP calls with failed records are rolled back. All the record changes of other calls are committed.

- **DuplicateRuleHeader**

  – **allowSave** (A Boolean value): if **allowSave** is set to true, duplicate records are saved. If this value is set to false duplicate records are prevented from being saved.

  – **includeRecordDetails** (A Boolean value): if **includeRecordDetails** is set to true, fields and values for records detected as duplicates are retrieved. If this value is set to false, only record IDs for records detected as duplicates are retrieved.

  – **runAsCurrentUser** (A Boolean value): if **runAsCurrentUser** is set to true, sharing rules for the current user are enforced when duplicate rules run. If this value is set to false, sharing rules specified in the class for the request are used.

    > If no sharing rules are specified, Apex code runs in system context and sharing rules for the current user are not enforced.

- **LocaleOptions**

  – **language** (A string value): specifies the language of the labels returned. The value must be a valid user locale (language and country), such as de_DE or en_GB. For more information about the locales, see *Force.com Web Service API Developer's Guide*.

- **DebuggingHeader**

  – **debugLevel** (A string value): specifies the level of detail in the debug header.

Input Items	Data Type	Description
		See *Apex Developer's Guide* for detailed information.

The response of the debugging information can be found in the SOAP message log. You might have to enable the debug role log for tracing the debugging errors.

- **PackageVersionHeader**

  Each **PackageVersion** includes the following elements:

  - **majorNumber** (An integer value): the major version number of a package version.

    A package version is denoted by *majorNumber.minorNumber*, for example 2.1.

  - **minorNumber** (An integer value): the minor version number of a package version.

    A package version is denoted by *majorNumber.minorNumber*, for example 2.1.

  - **namespace** (A string value): the unique namespace of the managed package.

- **EmailHeader**

  - **triggerAutoResponseEmail** (A Boolean value): specifies whether to trigger auto-response rules (true) or not (false), for leads and cases. In the Salesforce.com user interface, this email can be automatically triggered by a number of events, for example creating a case or resetting a user password.

  - **triggerOtherEmail** (A Boolean value): specifies whether to trigger email outside the organization (true) or not (false). In the Salesforce.com user interface, this email can be automatically triggered by creating, editing, or deleting a contact for a case.

  - **triggerUserEmail** (A Boolean value): specifies whether to trigger email that is sent to users in the organization (true) or not (false). In the Salesforce user interface, this email can be automatically triggered by a number of events: resetting a password, creating a new user, adding comments to a case, or creating or modifying a task.

- **OwnerChangeOptions**

  Each **OwnerChangeOption** includes the following elements:

  - **type** (A string value): represents the action performed or skipped, according to the given value for the **execute** field, when changing a record owner during an update call.

  - **execute** (A Boolean value): if it is set to true, the open activities of the record are transferred to the new record owner. If false, the original record owner retains the ownership.

    The default is false.

Output

On the **Output** tab, you can find the output value.

The following table lists the output elements on the **Output** tab of the Salesforce Update All activity:

Output Item		Data Type	Description
updateAllResponse			
result	**errors**	Complex	If errors occur during the activity, an array of error objects with the error code and description is returned.
	id	String	The ID of an sObject that you attempt to create.
	success	Boolean	Indicates whether the update activity succeeds (`true`) or not (`false`).

Fault

On the **Fault** tab, you can find the error code and error message of the Salesforce Update All activity. See Error Codes for more information about error codes and corrective actions to take.

Fault	Thrown when
SalesforceExecuteSOAPMethodException	An error occurred when calling an SOAP method. For example, wrong values are set in the Input fields or the Salesforce session timed out.
SalesforceLoginException	An error occurred when logging in to the Salesforce.com server.
SalesforceConnectionNotFoundException	An error occurred when the external session ID is not used or the Salesforce connection is not configured correctly.

Salesforce Upsert All

The Salesforce Upsert All activity creates new objects or updates existing objects; uses a custom field to determine the presence of existing objects.

This activity uses the external ID to determine whether it creates a new object or updates an existing one:

- If the external ID does not match, a new object is created.
- If the external ID is matched once, the existing object is updated.
- If the external ID is matched multiple times, an error is reported.
- When batch updating multiple objects where the external ID is the same for two or more objects in your batch call, those records are marked as errors in UpsertResult. The objects are neither created nor updated.

General

On the **General** tab, you can establish a connection to the Salesforce.com server.

The following table lists the configurations on the **General** tab of the Salesforce Upsert All activity:

Field	Module Property ?	Description
Name	No	The name displayed as the label of the activity in the process.
Salesforce Connection	Yes	The path to the Salesforce shared resource. Click the **Choose/Create Default Resource** icon 🔍 to select a usable connection for the activity.

Description

On the **Description** tab, you can enter a short description for the Salesforce Upsert All activity.

Input

On the **Input** tab, you can specify input values for the Salesforce Upsert All activity.

The following table lists the input elements on the **Input** tab of the Salesforce Upsert All activity:

Input Items	Data Type	Description
`connectionInfo` (All fields in this section are optional.)		
`serverUrl`	String	The web address of the endpoint that is used by this operation.
`sessionId`	String	The unique ID associated with this session.
`externalSessionIdUsed`	Boolean	Specifies whether an external session ID is used (`true`) or not (`false`). If the value is set to `true`, an external session ID is filled in the **sessionId** field. In this case, when the session has to be refreshed, the refreshed operation cannot be performed, and an exception is thrown.
`upsert`		
`batchSize`	Integer	Optional. When processing large amounts of data, the activity internally invokes several SOAP calls. This field is used to set the batch size for the number of records created or updated through an SOAP call. The value must be set between `1` and `200`. The default value is `200`. If the input value equals `-1`, it is set to the default value. If the input value is greater than `200`, it is set to `200`. If the input value is less than `1` (except `-1`), it is set to `1`.

Input Items	Data Type	Description
upsertSObjects		Each **upsertSObjects** includes the following elements: • **externalIDFieldName** (A string value): required. This field contains the name of the field defined as a key field in the sObject. • **sObjects** (A complex value): required. An array of one or more objects to create or update. This sObject can be converted to the specified type defined in the metadata schema, for example, the Salesforce_metadata, retrieved from the Salesforce.com server. You are not required to specify the **Id** field in the sObject. To create a new relationship between a newly created sObject (sObject A) and an existing sObject (sObject B), where the sObject A schema includes the sObject B schema, set the value of the key field in sObject B while upserting sObject A. If you modify the values of the fields in sObject B within the sObject A schema, the original sObject B is not modified. The key field in sObject B within the sObject A schema is only used to link sObject A with the original sObject B.
upsert_Optional (All fields in this section are optional.)		
_configData		Each **_configData** includes the following element: • **timeout** (An integer value): the timeout value specifies the number of milliseconds an internal API call waits before the data returns. The default value is 15000 milliseconds. You can also change the default timeout value by setting the properties. See Default Timeout Value Setting for detailed information.

Input Items	Data Type	Description
`headers`		Each `headers` includes the following elements:

- **`CallOptions`**

  - **`client`** (A string value): a string that identifies a particular client.

  - **`defaultNamespace`** (A string value): a string that identifies a developer namespace prefix. Use this field to resolve field names in managed packages without having to fully specify the fieldName everywhere.

- **`AssignmentRuleHeader`**

  - **`assignmentRuleId`** (A string value): the ID of a specific assignment rule to run for the case or the lead. It can be an inactive assignment rule. The ID can be retrieved by querying an AssignmentRule object. If specified, do not specify **`useDefaultRule`**. This element is ignored for Accounts, because all territory assignment rules are applied.

    The case and the lead are two predefined Salesforce.com schema types. For more information about these two types, see *Force.com Web Service API Developer's Guide*.

  - **`useDefaultRule`** (A Boolean value): if `true` for a case or a lead, uses the default (active) assignment rule for the case or the lead. If specified, do not specify an **`assignmentRuleId`** and type an empty string in the **`assignmentRuleId`** field. If `true` for an Account, all territory assignment rules are applied, and if `false`, no territory assignment rules are applied.

- **`MruHeader`**

  - **`updateMru`** (A Boolean value): specifies whether to update the list of most recently used items (`true`) or not (`false`), which is on the sidebar of the Salesforce.com user interface.

- **`AllowFieldTruncationHeader`**

  - **`allowFieldTruncation`** (A Boolean value): specifies whether to truncate field values that are too long (`true`) or not (`false`).

    Default value is `false`: no change in behavior. If a string or text area value is too large, the operation fails and the fault code `STRING_TOO_LONG` is returned.

- **`DisableFeedTrackingHeader`**

  - **`disableFeedTracking`** (A Boolean value): if **`disableFeedTracking`** is set to `true`, the changes made in the current call are not tracked in feeds.

    The default is `false`.

- **`StreamingEnabledHeader`**

Input Items	Data Type	Description

- **streamingEnabled** (A Boolean value): specifies whether you want to receive streaming notifications for changes to Salesforce data.

- **AllOrNoneHeader**

  - **allOrNone** (A Boolean value): if **allOrNone** is set to true, any failed records in a call cause all changes for the call to be rolled back. Record changes are not committed unless all records are processed successfully.

    The default value is false. Some records can be processed successfully while others are marked as failed in the call results.

    > When processing large amounts of data, the activity internally invokes several SOAP calls.
    >
    > However, this **AllOrNoneHeader** applies to each internal SOAP call individually. If **allOrNone** is set to true, only individual internal SOAP calls with failed records are rolled back. All the record changes of other calls are committed.

- **DuplicateRuleHeader**

  - **allowSave** (A Boolean value): if **allowSave** is set to true, duplicate records are saved. If this value is set to false duplicate records are prevented from being saved.

  - **includeRecordDetails** (A Boolean value): if **includeRecordDetails** is set to true, fields and values for records detected as duplicates are retrieved. If this value is set to false, only record IDs for records detected as duplicates are retrieved.

  - **runAsCurrentUser** (A Boolean value): if **runAsCurrentUser** is set to true, sharing rules for the current user are enforced when duplicate rules run. If this value is set to false, sharing rules specified in the class for the request are used.

    > If no sharing rules are specified, Apex code runs in system context and sharing rules for the current user are not enforced.

- **LocaleOptions**

  - **language** (A string value): specifies the language of the labels returned. The value must be a valid user locale (language and country), such as de_DE or en_GB. For more information about the locales, see *Force.com Web Service API Developer's Guide*.

- **DebuggingHeader**

  - **debugLevel** (A string value): specifies the level of detail in the debug header. See Salesforce.com document *Apex Developer's Guide* for detailed information.

Input Items	Data Type	Description
		The response of the debugging information can be found in the SOAP message log. To trace the debugging errors, the debug role log might be required.

- **PackageVersionHeader**

  Each **PackageVersion** includes the following nodes:

  - **majorNumber** (An integer value): the major version number of a package version.

    A package version is denoted by *majorNumber.minorNumber*, for example 2.1.

  - **minorNumber** (An integer value): the minor version number of a package version.

    A package version is denoted by *majorNumber.minorNumber*, for example 2.1.

  - **namespace** (A sting value): the unique namespace of the managed package.

- **EmailHeader**

  - **triggerAutoResponseEmail** (A Boolean value): specifies whether to trigger auto-response rules (`true`) or not (`false`), for leads and cases. In the Salesforce.com user interface, this email can be automatically triggered by a number of events, for example, resetting a user password.

  - **triggerOtherEmail** (A Boolean value): specifies whether to trigger email outside the organization (`true`) or not (`false`). In the Salesforce.com user interface, this email can be automatically triggered by creating, editing, or deleting a contact of a case.

  - **triggerUserEmail** (A Boolean value): specifies whether to trigger email that is sent to users in the organization (`true`) or not (`false`). In the Salesforce user interface, this email can be automatically triggered by a number of events: resetting a password, creating a new user, adding comments to a case, or creating or modifying a task.

- **OwnerChangeOptions**

  Each **OwnerChangeOption** includes the following elements:

  - **type** (A string value): represents the action performed or skipped, according to the given value for the **execute** field, when changing a record owner during an upsert call.

  - **execute** (A Boolean value): if it is set to `true`, the action represented by the type field is performed. If `false`, the action represented by the type field is skipped.

    The default is `false`.

Output

On the **Output** tab, you can find the output value.

The following table lists the output elements on the **Output** tab of the Salesforce Upsert All activity:

Output Item		Data Type	Description
upsertAllResponse			
result	**created**	Boolean	Indicates whether the record is created (`true`) or updated (`false`).
	errors	Complex	If errors occur during the activity, an array of error objects with the error code and description, is returned.
	id	String	The ID of an sObject that you attempt to create.
	success	Boolean	Indicates whether the Salesforce Create All activity has succeeded (`true`) or not (`false`).

Fault

On the **Fault** tab, you can find the error code and error message of the Salesforce Upsert All activity. See Error Codes for more information about error codes and corrective actions to take.

Fault	Thrown when
SalesforceExecuteSOAPMethodException	An error occurred when calling an SOAP method. For example, wrong values are set in the Input fields or the Salesforce session has timed out.
SalesforceLoginException	An error occurred when logging in to the Salesforce.com server.
SalesforceConnectionNotFoundException	An error occurred when the external session ID is not used or the Salesforce connection is not configured correctly.

Salesforce Outbound Message Listener

The SalesforceOutboundMessageListener process starter creates a process instance for incoming Salesforce outbound messages. Salesforce outbound messaging is part of the workflow rule functionality in Salesforce. It uses the notifications call to send SOAP messages over HTTP(S) to a designated endpoint when triggered by a workflow rule.

General

The following table lists the configurations on the **General** tab of the Salesforce Outbound Message Listener activity:

Field	Module Property?	Description
Name	No	The name for the activity in the process definition.
Http Connection	Yes	The path to the http connection resource containing the connection information of the Salesforce Outbound Message Listener activity. Click the **Choose/Create Default Resource** icon to select a usable connection for the activity.
WSDL	No	The file path to the Outbound Message WSDL file. To open the Select Salesforce WSDL window, click the **Select Outbound wsdl file from workspace** icon . Select the one WSDL file related to the current Salesforce Outbound Message Listener to apply to your activity. For the changes in the WSDL file to be reflected in the output section at run time, click the **Refresh** icon .
Manual Confirm	No	Enables you to manually acknowledge an event that is received by the outbound message listener activity. You can acknowledge the event after all the activities have processed the event or after any one of the activities in the process. You must add the Confirm activity after the listener in the process that has the SalesforceOutboundMessageListener activity.

If the Salesforce Outbound Message Listener activity is used in a process, the Endpoint URL of the process is http://`host:port`/URI , in which *host* and *port* are the values in the **Host** and **Port** fields of the HTTP Connector shared resource and URI is the value in the WSDL file. For example, the **Host** and **Port** fields in the HTTP Connector shared resource are `192.168.0.1` and `8443` , and the Endpoint URL in the WSDL file is `http://192.168.66.99:8090/outbound/test`, so the real Endpoint URL that plug-in uses to start the listener is `http://192.168.0.1:8443/outbound/test` .

Description

On the **Description** tab, you can enter a short description for the Salesforce Outbound Message Listener activity.

Advanced

The following table lists the configurations on the **Advanced** tab of the Salesforce Outbound Message Listener activity:

Field	Description
Sequence Key	This field contains an XPath Expression that specifies the order in which processes run. Process instances with sequencing keys evaluated as the same value are executed sequentially according to the order of creation.
Custom Job Id	This field contains an XPath expression that specifies a custom ID for the process instance.

Making Authenticated Web Service Callouts Using SSL

Callout is a useful feature with which you can connect to other web services and exchange data. You can use it to notify other services if data changes in your environment or you can retrieve data from a remote system.

Callouts can be secured by using SSL, in which the client and the server present certificates to prove identities.

The following topics describe two SSL authentications:

- One-Way SSL Authentication
- Two-Way SSL Authentication

One-Way SSL Authentication

Only the identity of the server is represented by digital certificates in the one-way SSL authentication.

For outbound messaging, one-way SSL is chosen by default when you configure HTTPS in Salesforce.com site. After you have defined an outbound message, you are required to configure the one-way SSL authentication:

Procedure

1. Log on to Salesforce.com and click **Build** > **Create** > **Workflow & Approvals** > **Outbound Messages** from the left panel.
   All outbound messages are displayed in the All Outbound Messages panel.

2. Click the one you want to use.
   The Workflow Outbound Message Detail panel is displayed.

3. Right-click **Click for WSDL**, next click **Save Link As**.

4. Type a file name or accept the default name to save the `.wsdl` file to your local directory.

5. Import the `.wsdl` file into your project.

6. Configure Keystore Provider.

7. Configure SSL Server.

8. Specify HTTP Connector and select the **Confidentiality** check box in the Security panel.

   > See *TIBCO ActiveMatrix BusinessWorks Bindings and Palettes Reference* on how to configure Keystore Provider, SSL Server and HTTP Connector.

9. Click and configure the **Salesforce Outbound Message Listener** starter in the Process editor:

   a) Click the **Choose/Create Default Resource** icon [🔍] to select the http connection you just created. Click **OK**.

   b) Click the **Select Outbound wsdl file from workspace** icon [📄] beside the **WSDL** field to select the `.wsdl` file imported. Click **OK**.

10. Click **File** > **Save**.

Two-Way SSL Authentication

The identities of the client and server are both represented by digital certificates in two-way SSL.

Currently, Salesforce supports both self-signed and CA-signed certificates. To configure the two-way SSL authentication, perform the following steps:

Procedure

1. Log on to Salesforce.com and click **Build** > **Create** > **Workflow & Approvals** > **Outbound Messages** from the left panel.
   All outbound messages are displayed in the **All Outbound Messages** panel.

2. Click the one you want to use.
   The Workflow Outbound Message Detail panel is displayed.

3. Right-click **Click for WSDL**, next click **Save Link As**.

4. Type a file name or accept the default name to save the `.wsdl` file to your local directory.

5. Copy the `.wsdl` file into your project.

6. Configure Keystore Provider.

7. Configure SSL Server.

   a) Select the **Enable Mutual Authentication** check box in the Basic SSL Server Configuration panel.
   b) Select **required** from the **Client Auth Type** list.

8. Specify HTTP Connector and select the **Confidentiality** check box in the Security panel.

   > See *TIBCO ActiveMatrix BusinessWorks Bindings and Palettes Reference* on how to configure Keystore Provider, SSL Server and HTTP Connector.

9. Click and configure the **Salesforce Outbound Message Listener** starter in the Process editor:

   a) Click the **Choose/Create Default Resource** icon 🔍 to select the http connection you just created. Click **OK**.

   b) Click the **Select Outbound wsdl file from workspace** icon 📄 beside the **WSDL** field to select the `.wsdl` file imported. Click **OK**.

10. Click **File** > **Save**.

Salesforce Subscriber

The Salesforce Subscriber (earlier known as Salesforce Topic Subscriber) activity subscribes to a single Salesforce streaming topic or event and generates a process event from each message received. To subscribe to a topic or event in Salesforce, you must configure the Salesforce Subscriber activity in TIBCO Business Studio .

> A process instance that contains the Salesforce Subscriber activity can be deployed on multiple hosts in an environment. Each of these process instances that is deployed on multiple hosts, acts as a separate subscriber and receives all the messages originating from the Push Topic, Platform Event, or Change Data Capture.

General

On the **General** tab, you can establish a connection to the Salesforce.com server. The following table lists the configurations on the **General** tab of the Salesforce Subscriber activity:

Field	Module Property ?	Description
Name	No	The name displayed as the label of the activity in the process.

Field	Module Property ?	Description
Salesforce Connection	Yes	The path to the Salesforce shared resource. Click the **Choose/Create Default Resource** icon 🔍 and select a usable connection for the activity.
Subscriber Type	No	The type of subscriber. You can select either Push Topic, Platform Event, or Change Data Capture. If you do not select any value for this field, the activity defaults to the Push Topic value. 📝 To migrate projects from plug-in version 6.6 or earlier to the current version without any error, use WSDL 46.0 or later. For more information about the Change Data Capture type, see the Salesforce Data Capture Developer Guide.
Salesforce Object	No	Ensures that the metadata for your development environment is up to date. It contains all the objects configured on the Salesforce system to which you are connected, including custom objects. If you leave it blank, the output is a JSON string. In this case the output schema is a text string containing the JSON representation of the object. However, if you select the object being streamed, its fields are displayed in the output. To subscribe to Change Data Capture for a particular sObject: Select the corresponding ChangeEvent object from the list. For example, to subscribe to Change Data Capture for Account object, select the AccountChangeEvent object. The channel and output sections are updated accordingly. 📝 When you add custom fields to a Platform Event after you subscribe to it, to see these new fields in the output of the activity, you must refresh the metadata and reselect the event from the Salesforce object list and save the project.
Channel	Yes	The topic name or event name of the stream to which the listener subscribes. The topic name starts with `/topic/` and the change data capture starts with `/data/` and the event name starts with `/event/` in the value and continues with the topic name or event name configured on the Salesforce server. 📝 You can configure this activity only for a topic or event that exists on the Salesforce server.
Filter String	Yes	The filter string that can be used to fetch specific details about the Salesforce Object through the channel. **Example:** `?Firstname=TestFilter` **Condition:** This feature is available only when the subscriber type is Push Topic.

Field	Module Property ?	Description
Persist ReplayID	Yes	When selected, the messages are replayed from the replay ID that is stored in the database.
		If there is no replay ID in the database, then it is replayed from the value provided in the **Replay From** field.
		To use this feature, you must configure either a custom database or engine persistence. If the Custom Database Configuration on the Salesforce Connection page is enabled, then the replay ID persists on the custom database configured or else it persists on the engine database.
		On the TIBCO BusinessWorks™ Container Edition engine, persistence is not supported. Therefore, you must configure custom database to use this feature.
		To use this feature, run the `create.sql` script to create the tables required to store the replay ID. The scripts for all databases are available at `<TIBCO_HOME>/bw/palettes/salesforce/<version>/dbscripts/`.
		To configure the TIBCO BusinessWorks engine persistence, see "Configuring Database for the Engine" topic in the *Application Development* guide in the ActiveMatrix BusinessWorks™ documentation.
		When the replay ID goes beyond the 24-hour window of Salesforce, it is no longer valid. In such a scenario, the Salesforce Subscriber activity tries to subscribe to the channel only once more by using the value provided in the **Replay From** field. However, the value in the **Replay From** field must be either -1 or -2.
		The Salesforce Subscriber activity also retires only once more in case of non-persistent subscribers if the replay ID has been in the memory for more than 24 hours (when the last event has been received 24 hours ago). The value in the **Replay From** field must be -1 or -2.
Replay From	Yes	Salesforce stores events for 24 hours. By configuring this field, you can retrieve events that are in the retention window.
		• -2 starts the replay from 24 hours ago.
		• -1 starts the replay from the moment the listener starts.
		• An actual replay ID means all events after that ID can be retrieved.
		For example, if the actual ID is 7, then replay starts from 8 onwards.

Description

On the **Description** tab, enter a short description for the Salesforce Subscriber activity.

Advanced

The following table lists the configurations on the **Advanced** tab of the Salesforce Subscriber activity:

Field	Description
Sequence Key	This field contains an XPath Expression that specifies the order in which processes run. Process instances with sequencing keys evaluated as the same value are run in the order they are created.
Custom Job Id	This field contains an XPath expression that specifies a custom ID for the process instance.

Output

On the **Output** tab, you can find the output value.

The following table lists the output elements on the **Output** tab of the Salesforce Subscriber activity:

Output Item		Data Type	Description
data			
event	**createdDate**	String	The date when the event was created.
	replayId	Integer	Indicates the replay ID of the event.
	type	String	Indicates the type of the event.
sObject		Complex	An array of sObjects representing the objects from Salesforce and containing data fetched from Salesforce. These sObjects can be substituted by using the derived objects of sObject defined in `Salesforce_metadata.xsd` in case of partner WSDL type or the enterprise WSDL in case of enterprise WSDL type.

For parsing according to the output schema defined as per releases prior to this release, set the property `com.tibco.bw.palette.salesorce.streaming.parseold` to true.

Salesforce Streaming API Properties

Salesforce Subscriber activity exposes the following system properties which you can use when sending requests, receiving responses, and parsing a response from the Salesforce channel:

Property	Description
`com.tibco.plugin.salesforce.streaming.buffer`	Sets the maximum size of the incoming message in megabytes. To avoid the buffering capacity exceed error, set this property to an appropriate value. However, if the Subscriber encounters a buffering capacity exceeded error, an exception is thrown. It can be caught in the CatchAll block and you can filter on the error code and take appropriate steps to recover. When the Subscriber encounters a buffering capacity exceeded error, an exception is thrown and the Subscriber disconnects and unsubscribes from the channel.
`com.tibco.plugin.salesforce.streaming.network.delay`	Sets the maximum time in milliseconds to wait before considering a request to the Bayeux server. To avoid connection timeout error, set this property to an appropriate value.

Salesforce Bulk API Palette

The Salesforce Bulk API palette is REST based. It uses the same log in module as the SOAP based Salesforce palette and uses the same shared resource. You can use the activities of this palette to insert, update, or delete large sets of data.

The Salesforce Bulk API palette includes the following activities:

- Salesforce Bulk Operation
- Salesforce Bulk Query
- Salesforce Check Status
- Salesforce Get Result
- Salesforce Get Subset Result
- Salesforce Close Job

Before using the Salesforce Bulk API palette, you must ensure that the Salesforce Connection shared resource is configured with valid credentials and the latest metadata is fetched in the project. For more information, see the Salesforce Tools topic.

Salesforce Bulk Operation

The Salesforce Bulk Operation activity performs operations such as insert, upsert, update, delete, hard delete on records in batches.

General

On the **General** tab, you can establish a connection to the Salesforce.com server.

The following table lists the configurations on the **General** tab of the Salesforce Bulk Operation activity:

Field	Module Property ?	Description
Name	No	The name displayed as the label of the activity in the process.
Salesforce Connection	Yes	The path to the Salesforce shared resource. Click the **Choose/Create Default Resource** icon ⌕ to select a usable connection for the activity.
Operation	Yes	Select one of the following operations from the drop-down list: • Insert: Insert records for an sObject. • Upsert: Update existing records; if the object ID is not present, it inserts a new record. • Update: Update the records. • Delete: Delete the records, but store them in Recycle Bin. • Hard delete: Delete the records from Recycle Bin too.

Field	Module Property?	Description
Format	Yes	Select one of the following format of the file types from the drop-down list: • CSV • XML • JSON • ROW • ZIP_XML • ZIP_CSV • ZIP_JSON 📝 ROW format allows you to add records on the **Input** tab. Other formats are provided for specific file types.
Object	No	Click **Pick Object** wizard to pick a Salesforce object on which the bulk operation needs to be performed.
Filename	Yes	Select a file of the format selected above. When you select ROW format on the **General** tab, the **Filename** option is disabled to substitute sObject on the **Input** tab.

Description

On the **Description** tab, you can enter a short description for the Salesforce Bulk Operation activity.

Advanced

On the **Advanced** tab, you can configure the following options of the Salesforce Bulk Operation activity.

The following table lists the configurations on the **Advanced** tab:

Field	Description
Field Mapper	Field Mapper wizard maps fields from user selected files to Salesforce object fields. Only one-to-one mapping from the provided file field to sObject is allowed. This option is available only for CSV, XML, or JSON file formats. For JSON and XML formats, Mapper refers only the first record in the file for field names. Ensure that the first record has all field names, even if they are empty. For CSV files the record structure is defined only for the first header line. 📝 If you do not use field mapper, Salesforce Bulk API palette uses file column name as field name. If no column names are provided, columns are mapped to the Salesforce object as it is.

Field	Description
Batch Size	Batch size is used to perform bulk operation in batches. The default value is `5000`. Maximum value allowed is `10000`.
Serial Processing	Select this option if you prefer to process batches serially. By default, Salesforce processes batches in parallel.

Input

On the **Input** tab, you can specify input values for the Salesforce Bulk Operation activity.

The following table lists the input elements on the **Input** tab of the Salesforce Bulk Operation activity:

Input Items	Data Type	Description
ConnectionInfo (All fields in this section are optional.)		
`ServerUrl`	String	The web address of the endpoint that is used by this operation.
`SessionId`	String	The unique ID associated with this session.
`ExternalSession IdUsed`	Boolean	Specifies whether an external session ID is used (`true`) or not (`false`).
		If the value is set to `true`, an external session ID is filled in the **sessionId** field.
		In this case, when the session has to be refreshed, the refreshed operation cannot be performed, and an exception is thrown.
Bulk (All fields in this section are optional.)		
`operation`	String	Specifies the operation to be invoked. You can enter one of these values: Insert, Update, Upsert, Delete, and Hard Delete.
`object`	String	Specifies the Salesforce object name for the activity operation.
`format`	String	Specifies the format of the records file upload for the operation to be invoked. You can enter one of these formats: CSV, JSON, XML, ZIP_XML, ZIP_CSV or ZIP_JSON.
`filename`	String	Specifies the absolute path of the file.
`batchSize`	Integer	Batch size is used to perform bulk operation in batches. The default value is `5000`. Maximum value allowed is `10000`.
`serialProcessin g`	Boolean	Specifies if serial processing is used (`true`) or not (`false`). If value is set to `false` then the Salesforce processes batches in parallel.
`externalIdField Name`	String	Optional. This field is used during upsert operation to provide the external ID.
		It contains the name of the field defined as a key field in the sObject.

Input Items	Data Type	Description
`sObjects`	Complex	Map your sObjects from this field.
BulkOptional (All fields in this section are optional.)		
`timeout`	Long	The timeout value specifies the number of milliseconds an internal API call waits before the data returns. **Default value:** Infinite (that is, activity times out only after all records are processed) If the timeout value provided is not sufficient, then only some batches are submitted, resulting in processing of partial records.
`Content-Type`	String	Specifies the format for your request and response. Content Type is applicable only for ROW format. On the **Input** tab of the Salesforce Bulk Operation activity, Content Type accepts only the following values: • "Json" or "application/json" for JSON format • "Csv" or "text/csv" for CSV format • "Xml" or "application/xml" for XML format For any other value, the Bulk request is sent in the default XML format.
`Sforce-Disable-Batch-Retry`	String	Specifies if retries are disabled for unfinished batches included in the operation. Field values are either `True` or `False`.
`Sforce-Line-Ending`	String	Specifies whether line endings are read as line feeds (LFs) or as carriage returns and line feeds (CRLFs).
`Sforce-Enable-PK-Chunking`	String	Enables automatic primary key (PK) chunking for a bulk query operation. This option is used for Bulk Query activity only.

Output

On the **Output** tab, you can find the output value.

The following table lists the output elements on the **Output** tab of the Salesforce Bulk Operation activity:

Output Item		Data Type	Description
SalesforceBulkOperationOutput			
`result`	`jobId`	String	The ID of the job that is submitted to the Salesforce server.
	`success`	Boolean	Indicates whether the job was successfully queued (`true`) or not (`false`).

Fault

On the **Fault** tab, you can find the error code and error message of the Salesforce Bulk Operation activity. See Error Codes for more information about error codes and corrective actions to take.

Fault	Thrown when..
SalesforceRest Exception	There is a problem with REST request, response or any other REST API error.
SalesforceLogi nException	An error occurred when logging in to the Salesforce.com server.
SalesforceConn ectionNotFoun dException	An error occurred when the external session ID was not used or the Salesforce connection was not configured correctly.

Salesforce Bulk Query

The Salesforce Bulk Query activity supports SOQL for Bulk API.

General

On the **General** tab, you can establish a connection to the Salesforce.com server.

The following table lists the configurations on the **General** tab of the Salesforce Bulk Query activity:

Field	Module Property ?	Description
Name	No	The name displayed as the label of the activity in the process.
Salesforce Connection	Yes	The path to the Salesforce shared resource. Click the **Choose/Create Default Resource** icon 🔍 to select a usable connection for the activity.
Query Type	Yes	Select a query type from the drop-down list. The default type is Query. If you select **Query** from the drop-down list, the operation queues a job for retrieving existing records only. If you select **QueryAll**, the operation queues a job for retrieving all archived and deleted records.
Format	Yes	Select one of the following format of the file types in which you want to view the Query Results on Salesforce server from the drop-down list: • XML • JSON • CSV

Field	Module Property ?	Description
Salesforce Query	No	Displays the query to be executed against the specified object. Click **Query Builder** to open the wizard to build your query or clear the existing query with **Clear Query**. For more information about the query builder, see Salesforce Query Builder.
Prepared Statement	No	If you use a prepared parameter '?' in the query, then define the parameter name and its data type in the statement. You can add multiple prepared parameters using the add button. The data type Byte is not supported for the parameters in the query.
Serial Processing	No	Select this option if you prefer to process batches serially. By default, Salesforce processes batches in parallel.

Description

On the **Description** tab, you can enter a short description for the Salesforce Bulk Query activity.

Input

On the **Input** tab, you can specify input values for the Salesforce Bulk Query activity.

The following table lists the input elements on the **Input** tab of the Salesforce Bulk Query activity:

Input Items	Data Type	Description
ConnectionInfo (All fields in this section are optional.)		
`ServerUrl`	String	The web address of the endpoint that is used by this operation.
`SessionId`	String	The unique ID associated with this session.
`ExternalSession IdUsed`	Boolean	Specifies whether an external session ID is used (`true`) or not (`false`). If the value is set to `true`, an external session ID is filled in the **sessionId** field. In this case, when the session has to be refreshed, the refreshed operation cannot be performed, and an exception is thrown.
Bulk (All fields in this section are optional.)		
`queryType`	String	Specifies the query type, that is, `Query` or `QueryAll`.

Input Items	Data Type	Description
`format`	String	Select one of the following format of the file types in which you want to view the Query Results on Salesforce server from the drop-down list: • XML • JSON • CSV
`serialProcessing`	Boolean	Specifies if serial processing is used (`true`) or not (`false`). If value is set to `false` then the Salesforce processes batches in parallel.
`query`	String	Specifies the query to be executed against the specified object. Query specified on the **Input** tab gets priority over query added on the **General** tab.
`fields`		This field is visible when you define a prepared parameter on the **General** tab. You can pass the prepared parameters values under the parameters.
BulkOptional (All fields in this section are optional.)		
`timeout`	Long	The timeout value specifies the number of milliseconds an internal API call waits before the data returns. If no timeout value is provided, then 180000 milliseconds is considered as default.
`Sforce-Disable-Batch-Retry`	String	Specifies if retries are disabled for unfinished batches included in the operation. **Valid values:** `True` or `False`
`Sforce-Line-Ending`	String	Specifies whether line endings are read as line feeds (LFs) or as carriage returns and line feeds (CRLFs).
`Sforce-Enable-PK-Chunking`	String	Enables automatic primary key (PK) chunking for a bulk query operation. For example, you can set the chunk size as follows: `chunkSize=1000`

Output

On the **Output** tab, you can find the output value.

The following table lists the output elements on the **Output** tab of the Salesforce Bulk Query activity:

Output Item		Data Type	Description
SalesforceBulkQueryOutput			
`result`	`jobId`	String	The ID of the job that is queued on the Salesforce server.
	`success`	Boolean	Indicates whether the job was successfully queued (`true`) or not (`false`).

Fault

On the **Fault** tab, you can find the error code and error message of the Salesforce Bulk Query activity. See Error Codes for more information about error codes and corrective actions to take.

Fault	Thrown when..
SalesforceRest Exception	There is a problem with REST request, response or any other REST API error.
SalesforceLogi nException	An error occurred when logging in to the Salesforce.com server.
SalesforceConn ectionNotFoun dException	An error occurred when the external session ID was not used or the Salesforce connection was not configured correctly.

Salesforce Check Status

The Salesforce Check Status activity performs a check on the jobs and retrieves their status.

General

On the **General** tab, you can establish a connection to the Salesforce.com server.

The following table lists the configurations on the **General** tab of the Salesforce Check Status activity:

Field	Module Property ?	Description
Name	No	The name displayed as the label of the activity in the process.
Salesforce Connection	Yes	The path to the Salesforce shared resource. Click the **Choose/Create Default Resource** icon 🔍 to select a usable connection for the activity.
Wait for completion	No	Select the check box to wait for the completion of the job and get the job status. **Recommendation for Bulk Operation with large amount of data:** When a large amount of data is processed during a Bulk Operation activity (added, updated, or deleted) and Bulk Queries return a large amount of data, it might take time to process the entire request. In such a scenario, select this option and give a sufficiently large timeout value to complete the operation.

Field	Module Property ?	Description
Timeout	Yes	The timeout value specifies the number of milliseconds an internal API call waits before returning the data. The field is editable only when you select the **Wait for completion** check box. **Default value:** 50000 milliseconds (that is, activity times out only after all records are processed) **If timeout value is not specified:** The system set the value to infinite (that is activity times out only after all records are processed) Data type for Module property of both **Timeout** and **Interval** should be Integer.
Interval	Yes	Specifies the time interval between each check status call. The field is editable only when you select the **Wait for completion** check box. **Default interval value:** 1000 milliseconds It is recommended that you choose this interval as per the submitted job for optimal API usage. Data type for Module property of both **Timeout** and **Interval** should be Integer.

Description

On the **Description** tab, you can enter a short description for the Salesforce Check Status activity.

Input

On the **Input** tab, you can specify input values for the Salesforce Check Status activity.

The following table lists the input elements on the **Input** tab of the Salesforce Check Status activity:

Input Items	Data Type	Description
ConnectionInfo (All fields in this section are optional.)		
`ServerUrl`	String	The web address of the endpoint that is used by this operation.
`SessionId`	String	The unique ID associated with this session.
`ExternalSession IdUsed`	Boolean	Specifies whether an external session ID is used (`true`) or not (`false`). If the value is set to `true`, an external session ID is filled in the **sessionId** field. In this case, when the session has to be refreshed, the refreshed operation cannot be performed, and an exception is thrown.
CheckStatus		

Input Items	Data Type	Description
`jobId`	String	Required. The ID of the job that is queued on the Salesforce server.
`timeout`	Integer	Optional. The timeout value specifies the number of milliseconds an internal API call waits before the data returns. **Default value:** Infinite
`waitForCompletion`	Boolean	Optional. Specifies a boolean value to wait for the completion of all jobs.
`interval`	Integer	Optional. Specifies the time interval between each check status call.
CheckStatusOptional (All fields in this section are optional.)		
`Sforce-Disable-Batch-Retry`	String	Specifies if retries are disabled for unfinished batches included in the operation. Field values are either `True` or `False`.
`Sforce-Line-Ending`	String	Specifies whether line endings are read as line feeds (LFs) or as carriage returns and line feeds (CRLFs).
`Sforce-Enable-PK-Chunking`	String	Enables automatic primary key (PK) chunking for a bulk query operation. This option is used for Bulk Query activity only.

Output

On the **Output** tab, you can find the output value.

The following table lists the output elements on the **Output** tab of the Salesforce Check Status activity:

Output Item	Description
SalesforceBulkQueryOutput	
`jobInfo`	Provides all information about the Job, for example the jobId, operation, object, createdById, status, apiVersion, and others.

Fault

On the **Fault** tab, you can find the error code and error message of the Salesforce Check Status activity. See Error Codes for more information about error codes and corrective actions to take.

Fault	Thrown when..
SalesforceRest Exception	There is a problem with REST request, response or any other REST API error.
SalesforceLogi nException	An error occurred when logging in to the Salesforce.com server.

Fault	Thrown when..
SalesforceConn ectionNotFoun dException	An error occurred when the external session ID was not used or the Salesforce connection was not configured correctly.

Salesforce Get Result

The Salesforce Get Result activity is used to retrieve the job results for Bulk Operation and Bulk Query.

General

On the **General** tab, you can establish a connection to the Salesforce.com server.

The following table lists the configurations on the **General** tab of the Salesforce Get Result activity:

Field	Module Property ?	Description
Name	No	The name displayed as the label of the activity in the process.
Salesforce Connection	Yes	The path to the Salesforce shared resource. Click the **Choose/Create Default Resource** icon 🔍 to select a usable connection for the activity.
Output Format	Yes	Select one of the following format of the file types from the drop-down list: • JSON • XML • CSV • ROW 📝 If you select ROW format during Bulk Operation and want the Get Result to be written into a file then you should select XML file type in Get Result
Output File	Yes	Specifies the absolute path for the Output file. The file format should be compatible with the format selected for Bulk Operation Activity.

Description

On the **Description** tab, you can enter a short description for the Salesforce Get Result activity.

Input

On the **Input** tab, you can specify input values for the Salesforce Get Result activity.

The following table lists the input elements on the **Input** tab of the Salesforce Get Result activity:

Input Items	Data Type	Description
GetResultConnection (All fields in this section are optional.)		
`ServerUrl`	String	The web address of the endpoint that is used by this operation.
`SessionId`	String	The unique ID associated with this session.
`ExternalSession IdUsed`	Boolean	Specifies whether an external session ID is used (`true`) or not (`false`).
		If the value is set to `true`, an external session ID is filled in the **sessionId** field.
		In this case, when the session has to be refreshed, the refreshed operation cannot be performed, and an exception is thrown.
GetResult		
`jobId`	String	Required. Specifies the job Id for which the result is to be obtained.
`fileLocation`	String	Optional. Specifies the file path of the records file. This field is not displayed when ROW format is selected on the **General** tab.
GetResultOptional (All fields in this section are optional.)		
`timeout`	Long	The timeout value specifies the number of milliseconds an internal API call waits before the data returns.
		Default value is set to infinite. If timeout is not provided it fails with `TimeoutError`.
`Sforce-Disable-Batch-Retry`	String	Specifies if retries are disabled for unfinished batches included in the operation. Field values are either `True` or `False`.
`Sforce-Line-Ending`	String	Specifies whether line endings are read as line feeds (LFs) or as carriage returns and line feeds (CRLFs).
`Sforce-Enable-PK-Chunking`	String	Enables automatic primary key (PK) chunking for a bulk query operation. This option is used for Bulk Query activity only.

Output

On the **Output** tab, you can find the output value.

The following table lists the output elements on the **Output** tab of the Salesforce Get Result activity:

Output Item		Data Type	Description
SalesforceGetResultOutput			
`result`	`jobId`	String	The ID of an sObject used later to check job status and get job result.
	`success`	Boolean	Indicates whether the bulk query activity has succeeded (`true`) or not (`false`).

Output Item	Data Type	Description
`filepath`	String	Indicates the file path of the results.
`unprocess edBatches`	Integer	Specifies the number of batches that are not processed.
`record`		The fields under `record` refer to the actual records for that Job ID. If you perform a Get Result operation after the Bulk operation, then the records display the record that was either inserted, upserted, updated, or deleted using bulk operation. If the Job ID is of a bulk query then it displays the records that were queried for the bulk query.

Fault

On the **Fault** tab, you can find the error code and error message of the Salesforce Get Result activity. See Error Codes for more information about error codes and corrective actions to take.

Fault	Thrown when..
SalesforceRest Exception	There is a problem with REST request, response or any other REST API error.
SalesforceLogi nException	An error occurred when logging in to the Salesforce.com server.
SalesforceConn ectionNotFoun dException	An error occurred when the external session ID was not used or the Salesforce connection was not configured correctly.

Salesforce Get Subset Result

The Salesforce Get Subset Result activity is used to retrieve the job results in subsets for a Bulk Query activity. When you want to use this activity's subset feature, it is recommended to use the activity in a Repeat loop with a condition for checking if the last batch has been processed.

If you use a Bulk Query with PK Chunking, it is advised that you use the Salesforce Check Status activity. Select the **Wait for Completion** option. Define a sufficiently large timeout value (to avoid loss of data).

General

On the **General** tab, establish a connection with the Salesforce.com server. The following table lists the configurations on the **General** tab of the Salesforce Get Subset Result activity:

Field	Module Property ?	Description
Name	No	Enter the name displayed as the label of the activity in the process.

Field	Module Property ?	Description
Salesforce Connection	Yes	Enter the path to the Salesforce shared resource. Click the **Choose/Create Default Resource** icon 🔍 to select a usable connection for the activity.
Output Format	Yes	Select a format from the list. 📝 If you select ROW format during Bulk Operation and want the Get Result to be written to a file, then select XML file type in Get Result.
Output File	Yes	Specify the absolute path for the Output file. The file format must be compatible with the format selected for the Bulk Operation activity.

Description

On the **Description** tab, enter a short description for the Salesforce Get Subset Result activity.

Input

On the **Input** tab, specify the information required to configure the Salesforce Get Subset Result activity.

Input Items	Data Type	Description
GetSubsetResultConnection (All fields in this section are optional.)		
`ServerUrl`	String	Enter the web address of the endpoint that is used by this operation.
`SessionId`	String	Enter the unique ID associated with this session.
`ExternalSession IdUsed`	Boolean	Specify whether an external session ID is used (`true`) or not (`false`). If you set this value to `true`, an external session ID is filled automatically in the **sessionId** field. 📝 If the ExternalSessionIdUsed is set to true, the session cannot be refreshed and any attempt to do so results in an exception.
GetSubsetResult		
`jobId`	String	Required. Specify the job ID for which the result is to be obtained.
`fileLocation`	String	Optional. Specify the file path to the records file. This field is not displayed when ROW format is selected on the **General** tab.
GetSubsetResultOptional (All fields in this section are optional.)		

Input Items	Data Type	Description
`timeout`	Long	Specify the number of milliseconds an internal API call waits before returning the data. **Default value:** Infinite (that is, activity times out only after all records are processed) **If timeout is not specified:** The activity fails with `TimeoutError`.
`Sforce-Disable-Batch-Retry`	String	Specifies if retries are disabled for unfinished batches included in the operation. **Valid values:**`True` or `False`.
`Sforce-Line-Ending`	String	Specify whether line endings are read as line feeds (LFs) or as carriage returns and line feeds (CRLFs).
`Sforce-Enable-PK-Chunking`	String	Enable automatic primary key (PK) chunking for a Bulk Query operation. **Condition:** This option is available only for Bulk Query activity.
`SubsetSetting`		
`subsetSize`	Integer	Specifies the number of batches to be fetched in one loop of the operation. **When this value is 0 or not specified:** Batches are not processed in subsets **Condition:** Set this property only if you have selected **ROW** as the output format.

Output

On the **Output** tab, you can find the output value.

The following table lists the output elements on the **Output** tab of the Salesforce Get Result activity:

Output Item		Data Type	Description
SalesforceGetSubsetResultOutput			
`result`	`jobId`	String	The ID of an sObject which can be used later to check the job status and get the job result.
	`success`	Boolean	Indicates whether the Bulk Query activity has succeeded (`true`) or not (`false`).
	`filepath`	String	Displays the file path to the results.
	`unprocessedBatches`	Integer	Specifies the number of batches that are not processed.

Output Item	Data Type	Description
record		The fields under **record** are the actual records for that Job ID. **Example:** If you perform a Get Subset Result operation after the Bulk Operation, then the records that were inserted, upserted, updated, or deleted by using Bulk Operation are displayed. If the Job ID is of a Bulk Query activity, then it displays the records that were queried. In the Salesforce Bulk Query activity results, the sObject element in a record can be coerced to 1. Partner WSDL elements in `Salesforce_Metadata.xsd` 2. Enterprise WSDL elements derived from sObject in the WSDL
isLastBatch	Boolean	`true` if no: Indicates that there are no more batches to be retrieved.

Fault

On the **Fault** tab, you can find the error code and error message of the Salesforce Get Subset Result activity. See Error Codes for more information about error codes and corrective actions to take.

Fault	Thrown when..
SalesforceRestException	There is a problem with REST request, response, or any other REST API error.
SalesforceLoginException	An error occurs while logging in to the Salesforce.com server.
SalesforceConnectionNotFoundException	An error occurs when the external session ID is not used or the Salesforce connection is not configured correctly.

Salesforce Close Job

The Salesforce Close Job activity is used to close the open jobs that are created by using the plug-in or by using external REST API invocations. This activity is especially useful when closing Bulk Query jobs with PK Chunking.

Do not use the Close Job activity in combination with Bulk Operation or Bulk Query without PK Chunking.

General

On the **General** tab, establish a connection with the Salesforce.com server.

The following table lists the configurations on the **General** tab of the Salesforce Close Job activity:

Field	Module Property ?	Description
Name	No	The name displayed as the label of the activity in the process.
Salesforce Connection	Yes	The path to the Salesforce shared resource. Click the **Choose/Create Default Resource** icon 🔍 to browse and select a connection.

Description

On the **Description** tab, enter a short description for the Salesforce Close Job activity.

Input

On the **Input** tab, specify the input values for the Salesforce Close Job activity.

The following table lists the input elements on the **Input** tab of the Salesforce Close Job activity:

Input Items	Data Type	Description
CloseJobConnection (All fields in this section are optional.)		
`ServerUrl`	String	Enter the web address of the endpoint that is used by this operation.
`SessionId`	String	Enter the unique ID associated with this session.
`ExternalSession IdUsed`	Boolean	Specify whether an external session ID is used (`true`) or not (`false`). If the value is set to `true`, an external session ID is automatically filled in the **sessionId** field. If the ExternalSessionIdUsed is set to true, the session cannot be refreshed and any attempt to do so results in an exception.
CloseJob		
`jobId`	String	Enter the ID of the bulk job that is going to be closed or aborted.
`state`	String	Enter the state to update the job to. **Valid values:** `Closed` or `Aborted` referring to either closing a job or aborting it.
`CloseJobOptional`		
`timeout`	Long	Specify the number of milliseconds an internal API call waits before returning the data. **Default value:** Infinite (that is, activity times out only after all records are processed)

Output

On the **Output** tab, you can find the output value.

The following table lists the output elements on the **Output** tab of the Salesforce Close Job activity:

Output Item		Data Type	Description
SalesforceCloseJobOutput			
CloseJobResult	jobId	String	Unique ID for this job as seen on Salesforce backend.
	operation	String	The processing operation for all the batches in the job. For example, update, create, delete, and so on.
	object	String	The object type for the data being processed.
	createdById	String	Salesforce ID of the user who created this job.
	createdByDate	String	The date and time in the UTC time zone when the job was created.
	systemModstamp	String	The date and time in the UTC time zone when the job was finished.
	state	String	The current state of processing for the job. It can be either Closed or Aborted depending on the value given as an input to the activity.
	concurrencyMode	String	The mode of processing the batches. It can be either serially or in parallel.
	contentType	String	The content type for the job. The values can be: CSV, XML, JSON. ZIP_CSV, ZIP_XML, ZIP_JSON
	numberBatchesQueued	Int	The number of batches queued for this job.
	numberBatchesInProgress	Int	The number of batches that are in progress for this job.
	numberBatchesCompleted	Int	The number of batches that have been completed for this job.
	numberBatchesFailed	Int	The number of batches that have failed for this job.
	numberBatchesTotal	Int	The number of batches submitted for processing in this job.

Output Item	Data Type	Description
numberRecords Processed	Int	The number of records already processed. This number increases as more batches are processed.
numberRetries	Int	The number of times that Salesforce has attempted to save the result of an operation.
apiVersion	String	The API version of the job set in the URI when the job was created.
numberRecords Failed	Int	The number of records that were not processed successfully in this job.
totalProcessing Time	Int	The time in milliseconds taken to process the job. It is the sum of the total processing time for all batches in the job.
apiActiveProce ssingTime	Int	The time in milliseconds taken to actively process the job and includes *apexProcessingTime*. However, it does not include the time that the job was in queue to be processed or the time required for serialization or deserialization.
apexProcessing Time	Int	The time in milliseconds taken to process triggers and other processes related to the job.

Fault

On the **Fault** tab, you can find the error code and error message of the Salesforce Close Job activity. See Error Codes for more information about error codes and corrective actions to take.

Fault	Thrown when..
SalesforceRestException	There is a problem with REST request, response or any other REST API error.
SalesforceLoginException	An error occurs when logging in to the Salesforce.com server.
SalesforceConnectionNotFoundException	An error occurs when the external session ID is not used or the Salesforce connection is not configured correctly.

Salesforce Bulk API 2.0 Palette

The Salesforce Bulk API 2.0 palette is REST based. It uses the same login module as the SOAP based Salesforce palette and uses the same shared resource. You can use the activities of this palette to insert, update, delete or query large sets of data. In Salesforce Bulk API 2.0, all data is uploaded as a single batch. The palette activities are based on Salesforce Bulk API version 2.0.

The Salesforce Bulk API 2.0 palette includes the following activities:

- Salesforce Bulk Ingest Operation
- Salesforce Bulk Ingest Query Job
- Salesforce Check Ingest Job Status
- Salesforce Get Ingest Job Result
- Salesforce Close Ingest Job

Before using the Salesforce Bulk API 2.0 palette, you must ensure that the Salesforce Connection shared resource is configured with valid credentials and the latest metadata is fetched in the project. For more information, see the Salesforce Tools topic.

Salesforce Bulk Ingest Operation Activity

The Salesforce Bulk Ingest Operation activity performs operations such as insert, upsert, update, delete, and hard delete on records provided through the **Input** tab or a file input and uploads all records to Salesforce as a single batch.

Because all records are uploaded as a single batch, all the records are taken in memory at the same time; therefore you must configure Java memory accordingly.

General

On the **General** tab, you can establish a connection to the Salesforce.com server.

The following table lists the configurations on the **General** tab of the Salesforce Bulk Ingest Operation activity:

Field	Module Property ?	Description
Name	No	The name displayed as the label of the activity in the process.
Salesforce Connection	Yes	The path to the Salesforce shared resource. Click the **Choose/Create Default Resource** icon 🔍 to select a usable connection for the activity.

Field	Module Property?	Description
Operation	Yes	Select one of the following operations from the drop-down list: • Insert: Insert records for an sObject. • Upsert: Update existing records; if the object ID is not present, it inserts a new record. • Update: Update the records. • Delete: Delete the records, but store them in Recycle Bin. • Hard delete: Delete the records from Recycle Bin too.
Format	Yes	Select one of the formats of the file types from the drop-down list. Options available are CSV and Row. You can use the Row format to add records on the **Input** tab. Other formats are provided for specific file types.
Line Ending	Yes	Specifies whether line endings are read as line feeds (LFs) or as carriage returns and line feeds (CRLFs).
CSV Separator	Yes	Select a separator as required. The options available are Comma, Backquote, Caret, Pipe, Semicolon, and Tab. You must select the same separator as that used in the CSV datafile you provide for the operation.
Object	No	Click the Pick Object wizard to pick a Salesforce object on which the bulk operation needs to be performed. Available only when you select the CSV format.
Filename	Yes	Select a file of the format selected above. Available only when CSV format is selected on the **General** tab.

Description

On the **Description** tab, you can enter a short description for the Salesforce Bulk Ingest Operation activity.

Advanced

On the **Advanced** tab, you can configure the following options of the Salesforce Bulk Ingest Operation activity.

The following table lists the configurations on the **Advanced** tab:

Field	Description
Field Mapper	Field Mapper wizard maps fields from user selected files to Salesforce object fields. Only one-to-one mapping from provided file field to sObject is allowed. This option is available only for CSV file format. For CSV files the record structure is defined only for the first header line. If you do not use field mapper, Salesforce Bulk API 2.0 palette uses file column name as field name. If no column names are provided, columns are mapped to the Salesforce object as it is.

Input

On the **Input** tab, you can specify input values for the Salesforce Bulk Ingest Operation activity.

The following table lists the input elements on the **Input** tab of the Salesforce Bulk Ingest Operation activity:

Input Items	Data Type	Description
ConnectionInfo (All fields in this section are optional.)		
`ServerUrl`	String	The web address of the endpoint that is used by this operation.
`SessionId`	String	The unique ID associated with this session.
`ExternalSession IdUsed`	Boolean	Specifies whether an external session ID is used (`true`) or not (`false`). If the value is set to `true`, an external session ID is filled in the **sessionId** field. In this case, when the session has to be refreshed, the refreshed operation cannot be performed, and an exception is thrown.
Bulk (All fields in this section are optional.)		
`operation`	String	Specifies the operation to be invoked. You can enter one of these values: Insert, Update, Upsert, Delete, and Hard Delete.
`object`	String	Specifies the Salesforce object name for the activity operation.
`format`	String	Specifies the format of the records file upload for the operation to be invoked. The formats available are CSV and Row.
`filename`	String	Specifies the absolute path of the file.
`lineEnding`	String	Specifies whether line endings are read as line feeds (LFs) or as carriage returns and line feeds (CRLFs).
`csvSeparator`	String	Specifies the CSV separator that is used in the file. You can select any one of these separators: Comma, Backquote, Caret, Pipe, Semicolon, and Tab.

Input Items	Data Type	Description
`externalIdField Name`	String	Optional. This field is used during upsert operation to provide the external ID.
		It contains the name of the field defined as an indexing field in the sObject.
`sObjects`	Complex	Substitute and map your sObjects from this field.
		Available only when Row format is selected.
BulkOptional		
`timeout`	Long	Specifies the number of milliseconds the activity waits before timing out.**Default value:** Infinite (that is, activity times out only after all records are processed)
		If the timeout value provided is not sufficient, the batch might not be uploaded to Salesforce.

Output

On the **Output** tab, you can find the output value.

The following table lists the output elements on the **Output** tab of the Salesforce Bulk Ingest Operation activity:

Output Item		Data Type	Description
SalesforceBulkOperationOutput			
jobInfo	`apiVersion`	String	The salesforce API version in which the job was created.
	`columnDelimiter`	String	Column Delimiter (symbol used to separate fields in the CSV records) used in the CSV job data. Values include COMMA, CARET, PIPE, BACKQUOTE, SEMICOLON, and TAB.
	`concurrencyMode`	String	For future use. Currently only parallel mode is supported in Bulk API 2.0.
	`contentType`	String	The format of the data being processed. Currently only CSV is supported for Bulk API 2.0.
	`contentUrl`	String	The URL to use for Upload Job requests for this job. Only provided if the job is in Open state.
	`createdById`	String	The ID of the user who created the job. The batch is created by using the same user.

Output Item		Data Type	Description
	createdDate	DateTime	The date and time in the UTC time zone when the job was created.
	externalIdFieldname	String	Optional. This field contains the name of the field defined as the key indexing field in the Salesforce object. It is used with the upsert operation.
	id	String	Unique ID of the job. The same can be mapped to further activities such as CheckStatus, GetResult, and CloseJob.
	jobType	String	The job's type. Values include: • BigObjectIngest: BigObjects job • Classic: Bulk API 1.0 job • V2Ingest: Bulk API 2.0 job
	lineEnding	String	The line ending used for CSV job data. This output field can be mapped to the GetResult activity.
	object	String	The Salesforce object type for the data being processed.
	operation	String	The processing operation specified. This can be mapped to further activities such as CheckStatus and GetResult.
	state	String	The current state of processing for the job. The BulkOperation and BulkQuery activities always update the job's state to UploadComplete. Other states are Open, Aborted, JobComplete, and Failed.
	systemModstamp	String	Date and time in the UTC time zone when the job finished.

Fault

On the **Fault** tab, you can find the error code and error message of the Salesforce Bulk Operation activity. See Error Codes for more information about error codes and corrective actions to take.

Fault	Thrown when..
SalesforceRest Exception	There is a problem with REST request, response or any other REST API error.

Fault	Thrown when..
SalesforceLoginException	An error occurred when logging in to the Salesforce.com server.
SalesforceBulk Exception	An error occurred when validating input parameters, requests, and parsing data before sending a request to Salesforce.

Salesforce Bulk Ingest Query Activity

The Salesforce Bulk Ingest Query supports SOQL to query large data for Bulk API 2.0

General

On the **General** tab, you can establish a connection to the Salesforce.com server.

The following table lists the configurations on the **General** tab of the Salesforce Bulk Ingest Query activity:

Field	Module Property ?	Description
Name	No	The name displayed as the label of the activity in the process.
Salesforce Connection	Yes	The path to the Salesforce shared resource. Click the **Choose/Create Default Resource** icon to select a usable connection for the activity.
Query Type	Yes	Select a query type from the drop-down list. The default type is Query. If you select **Query** from the drop-down list, the operation queues a job for retrieving existing records only. If you select **QueryAll**, the operation queues a job for retrieving all archived and deleted records.
Format	Yes	Select one of the following format of the file types in which you want to view the Query Results on Salesforce server from the drop-down list. The available format is CSV.
Query	No	Displays the query to be executed against the specified object. Click **Query Builder** to open the wizard to build your query or clear the existing query with **Clear Query**. For more information about the query builder, see Salesforce Query Builder.
Prepared Statement	No	If you use a prepared parameter '?' in the query, then define the parameter name and its data type in the statement. You can add multiple prepared parameters using the add button. The data type Byte is not supported for the parameters in the query.
Line Ending	Yes	Specifies whether line endings are read as line feeds (LFs) or as carriage returns and line feeds (CRLFs).

Field	Module Property ?	Description
CSV Separator	Yes	Specifies the CSV separator that is used in the file. You can select any one of these separators: Comma, Backquote, Caret, Pipe, Semicolon, and Tab.

Description

On the **Description** tab, you can enter a short description for the Salesforce Bulk Ingest Query activity.

Input

On the **Input** tab, you can specify input values for the Salesforce Bulk Ingest Query activity.

The following table lists the input elements on the **Input** tab of the Salesforce Bulk Ingest Query activity:

Input Items	Data Type	Description
ConnectionInfo (All fields in this section are optional.)		
`ServerUrl`	String	The web address of the endpoint that is used by this operation.
`SessionId`	String	The unique ID associated with this session.
`ExternalSession IdUsed`	Boolean	Specifies whether an external session ID is used (`true`) or not (`false`). If the value is set to `true`, an external session ID is filled in the **sessionId** field. In this case, when the session has to be refreshed, the refreshed operation cannot be performed, and an exception is thrown.
Bulk (All fields in this section are optional.)		
`queryType`	String	Specifies the query type, that is, `Query` or `QueryAll`.
`format`	String	The format of the file type in which you want to view the Query Results on Salesforce server. The file format supported is CSV.
`query`	String	Specifies the query to be executed against the specified object. Query specified on the **Input** tab gets priority over query added on the **General** tab.
BulkOptional		
`timeout`	Long	The timeout value specifies the number of milliseconds an activity waits before it times out. **Default value:** 18000 milliseconds **If timeout value is not specified:** The system sets the value to infinite (that is, activity times out only after all records are processed)

Output

On the **Output** tab, you can find the output value.

The following table lists the output elements on the **Output** tab of the Salesforce Bulk Ingest Query activity:

Output Item		Data Type	Description
SalesforceBulkQueryOutput			
jobInfo	`apiVersion`	String	The salesforce API version in which the job was created.
	`columnDelimiter`	String	Column Delimiter (symbol used to separate fields in the CSV records) used in the CSV job data. Values include COMMA, CARET, PIPE, BACKQUOTE, SEMICOLON, and TAB.
	`concurrencyMode`	String	For future use. Currently only parallel mode is supported in Bulk API 2.0.
	`contentType`	String	The format of the data being processed. Currently only CSV is supported for Bulk API 2.0.
	`contentUrl`	String	The URL to use for Upload Job requests for this job. Only provided if the job is in Open state.
	`createdById`	String	The ID of the user who created the job. The batch is created by using the same user.
	`createdDate`	DateTime	The date and time in the UTC time zone when the job was created.
	`externalIdFieldname`	String	Optional. This field contains the name of the field defined as the key indexing field in the Salesforce object. It is used with the upsert operation.
	`id`	String	Unique ID of the job. The same can be mapped to further activities such as CheckStatus, GetResult, and CloseJob.
	`jobType`	String	The job's type. Values include: • BigObjectIngest: BigObjects job • Classic: Bulk API 1.0 job • V2Ingest: Bulk API 2.0 job
	`lineEnding`	String	The line ending used for CSV job data. This output field can be mapped to the GetResult activity.

Output Item	Data Type	Description
`object`	String	The Salesforce object type for the data being processed.
`operation`	String	The processing operation specified. This can be mapped to further activities such as CheckStatus and GetResult.
`state`	String	The current state of processing for the job. The BulkOperation and BulkQuery activities always update the job's state to UploadComplete. Other states are Open, Aborted, JobComplete, and Failed.
`systemModstamp`	String	Date and time in the UTC time zone when the job finished.

Fault

On the **Fault** tab, you can find the error code and error message of the Salesforce Bulk Ingest Query activity. See Error Codes for more information about error codes and corrective actions to take.

Fault	Thrown when..
Salesforce RestException	There is a problem with REST request, response or any other REST API error.
Salesforce LoginException	An error occurred when logging in to the Salesforce.com server.
Salesforce BulkException	An error occurred when validating input parameters, requests, and parsing data before sending a request to Salesforce.

Salesforce Check Ingest Job Status Activity

The Salesforce Check Ingest Job Status activity performs a check on the jobs and retrieves their status.

General

On the **General** tab, you can establish a connection to the Salesforce.com server.

The following table lists the configurations on the **General** tab of the Salesforce Check Ingest Job Status activity:

Field	Module Property ?	Description
Name	No	The name displayed as the label of the activity in the process.

Field	Module Property ?	Description
Salesforce Connection	Yes	The path to the Salesforce shared resource. Click the **Choose/Create Default Resource** icon to select a usable connection for the activity.
Wait for completion	No	Select the check box to wait for the completion of the job and get the job status. **Recommendation for Bulk Operation with large amount of data:** When a large amount of data is processed during a Bulk Operation activity (added, updated, or deleted) and Bulk Queries return a large amount of data, it might take time to process the entire request. In such a scenario, select this option and give a sufficiently large timeout value to complete the operation.
Timeout	Yes	The timeout value specifies the number of milliseconds an activity waits before timing out. The activity will either timeout after given milliseconds or if the **Wait for completion** check box is selected, it will wait for the job to be completed, whichever is earlier. **Default value:** 50000 milliseconds. The activity adds a buffer of 5000 milliseconds to the given value to allow for processing time. For example, if the value is given as 50000 milliseconds then the activity will timeout after 55000 milliseconds. But the job check status poll is done only for the first 50000 milliseconds. (This does not apply for infinite timeout.) **If timeout value is not specified:** The system sets the value to infinite (that is, activity times out only after all records are processed) Data type for Module property of **Timeout** should be Integer.
Interval	Yes	Specifies the time interval between each check status call. The field is editable only when you select the **Wait for completion** check box. **Default interval value:** 1000 milliseconds It is recommended that you choose this interval as per the submitted job for optimal API usage. Data type for Module property of **Interval** should be Integer.

Description

On the **Description** tab, you can enter a short description for the Salesforce Check Ingest Job Status activity.

Input

On the **Input** tab, you can specify input values for the Salesforce Check Ingest Job Status activity.

The following table lists the input elements on the **Input** tab of the Salesforce Check Ingest Job Status activity:

Input Items	Data Type	Description
CheckStatusConnection (All fields in this section are optional.)		
`ServerUrl`	String	The web address of the endpoint that is used by this operation.
`SessionId`	String	The unique ID associated with this session.
`ExternalSession IdUsed`	Boolean	Specifies whether an external session ID is used (`true`) or not (`false`). If the value is set to `true`, an external session ID is filled in the **sessionId** field. In this case, when the session has to be refreshed, the refreshed operation cannot be performed, and an exception is thrown.
CheckStatus		
`jobId`	String	Required. The ID of the job that is queued on the Salesforce server.
`timeout`	Integer	Optional. The timeout value specifies the number of milliseconds an internal API call waits before the data returns.
`waitForCompleti on`	Boolean	Optional. Specifies a boolean value to wait for the completion of all jobs.
`interval`	Integer	Optional. Specifies the time interval between each check status call.
`operationType`	String	Supported operation types are insert, upsert, update, delete, harddelete, query, and queryall. It can be mapped from output of previous Bulk Ingest, Query, or Close Job activity.

It is recommended to provide all inputs (waitForCompletion, timeout, and interval) either from the **General** tab or from the **Input** tab.

Output

On the **Output** tab, you can find the output value.

The following table lists the output elements on the **Output** tab of the Salesforce Check Ingest Job Status activity:

Output Item		Data Type	Description
SalesforceBulkQueryCheckStatus			
jobInfo	`apiVersion`	String	The salesforce API version in which the job was created.

Output Item		Data Type	Description
	columnDelimiter	String	Column Delimiter (symbol used to separate fields in the CSV records) used in the CSV job data. Values include COMMA, CARET, PIPE, BACKQUOTE, SEMICOLON, and TAB.
	concurrencyMode	String	For future use. Currently only parallel mode is supported in Bulk API 2.0.
	contentType	String	The format of the data being processed. Currently only CSV is supported for Bulk API 2.0.
	contentUrl	String	The URL to use for Upload Job requests for this job. Only provided if the job is in Open state.
	createdById	String	The ID of the user who created the job. The batch is created by using the same user.
	createdDate	DateTime	The date and time in the UTC time zone when the job was created.
	externalIdFieldname	String	Optional. This field contains the name of the field defined as the key indexing field in the Salesforce object. It is used with the upsert operation.
	id	String	Unique ID of the job. The same can be mapped to further activities such as CheckStatus, GetResult, and CloseJob.
	jobType	String	The job's type. Values include: • BigObjectIngest: BigObjects job • Classic: Bulk API 1.0 job • V2Ingest: Bulk API 2.0 job
	lineEnding	String	The line ending used for CSV job data. This output field can be mapped to the GetResult activity.
	object	String	The Salesforce object type for the data being processed.
	operation	String	The processing operation specified. This can be mapped to further activities such as CheckStatus and GetResult.

Output Item	Data Type	Description
state	String	The current state of processing for the job. The BulkOperation and BulkQuery activities always update the job's state to UploadComplete. Other states are Open, Aborted, JobComplete, and Failed.
systemModstamp	String	Date and time in the UTC time zone when the job finished.

Fault

On the **Fault** tab, you can find the error code and error message of the Salesforce Check Ingest Job Status activity. See Error Codes for more information about error codes and corrective actions to take.

Fault	Thrown when..
SalesforceRestException	There is a problem with REST request, response or any other REST API error.
SalesforceLoginException	An error occurred when logging in to the Salesforce.com server.
SalesforceBulkException	An error occurred when validating input parameters, requests, and parsing data before sending a request to Salesforce.

Salesforce Get Ingest Job Result Activity

The Salesforce Get Ingest Job Result activity is used to retrieve completed job results for Bulk Operation and Bulk Query jobs created by using Bulk API 2.0.

General

On the **General** tab, you can establish a connection to the Salesforce.com server.

The following table lists the configurations on the **General** tab of the Salesforce Get Ingest Job Result activity:

Field	Module Property ?	Description
Name	No	The name displayed as the label of the activity in the process.
Salesforce Connection	Yes	The path to the Salesforce shared resource. Click the **Choose/Create Default Resource** icon 🔍 to select a usable connection for the activity.

Field	Module Property ?	Description
Operation Type	Yes	Type of operation to perform. Operation types available are insert, upsert, update, delete, hardDelete, query, queryAll.
Output Format	Yes	Select one of the following format of the file types from the drop-down list: CSV or Row
Result Type	Yes	The result of the operation. It can be success, failure, or unprocessed. Available only when BulkOperation is selected in Operation Type.
Output Filename	Yes	Specifies the absolute path for the Output file. The file format should be compatible with the format selected for Bulk Operation Activity. Available only when BulkOperation is selected in Operation Type.

Description

On the **Description** tab, you can enter a short description for the Salesforce Get Ingest Job Result activity.

Input

On the **Input** tab, you can specify input values for the Salesforce Get Ingest Job Result activity.

The following table lists the input elements on the **Input** tab of the Salesforce Get Ingest Job Result activity:

Input Items	Data Type	Description
GetSubsetResultConnection (All fields in this section are optional.)		
`ServerUrl`	String	The web address of the endpoint that is used by this operation.
`SessionId`	String	The unique ID associated with this session.
`ExternalSession IdUsed`	Boolean	Specifies whether an external session ID is used (`true`) or not (`false`). If the value is set to `true`, an external session ID is filled in the **sessionId** field. In this case, when the session has to be refreshed, the refreshed operation cannot be performed, and an exception is thrown.
GetSubsetResult		
`jobId`	String	Required. Specifies the job Id for which the result is to be obtained.
`filename`	String	Absolute path. Available only when CSV format is selected on the **General** tab.

Input Items	Data Type	Description
rolloverFileSize	Long	The rolloverFileSize denotes the maximum data size that can be written to the file before the writer rolls over to a new file. The new file is created in pattern named originalFilename_<number>.csv, where number is the consecutive number of files to which data is written.
		For example, by using GetResult activity if fetched records are of 20 Mb size and rollover size is provided as 2 Mb than records are saved in 10 different files (data, data_1, data_2 ....data_9) with max size of each file as 2 Mb.
operationType	String	Specifies the type of operation - insert, upsert, update, deleted, hardDelete, query, or queryAll.
resultType	String	Specifies whether the operation was successful, failure, or unprocessed.
csvSeparator	String	Specifies the CSV separator that is used in the file. You can select any one of these separators: Comma, Backquote, Caret, Pipe, Semicolon, and Tab.
LineEnding	String	Specifies whether line endings are read as line feeds (LFs) or as carriage returns and line feeds (CRLFs).
timeout	Long	Specifies the number of milliseconds the activity waits before timing out. If timeout is not provided there is no error, default is infinite.
QuerySubsetInfo		
maxRecords	Long	Specifies the number of records to be fetched in one loop of the operation. It should be used with a Repeat control group ideally with a check on isLastBatch.
		This is applicable to Query operations only.

The operationType, csvseparator, and lineEnding fields should be ideally mapped from previous ingest bulk or query operation.

Output

On the **Output** tab, you can find the output value.

The following table lists the output elements on the **Output** tab of the Salesforce Get Ingest Job Result activity:

Output Item		Data Type	Description
SalesforceGetResultOutput			
result	jobId	String	The ID of an sObject used later to check job status and get job result.

Output Item	Data Type	Description
`outputFilepath`	String	Outputs all the names of the files parse to be written to an array, but does not guarantee sequence. Available only when Format selected on the **General** tab is CSV.
`maxRecords`	Long	Specifies the maximum number of records fetched in a single loop iteration.
`locator`	String	Specifies the locator of the last read record received from Salesforce in case all records are not read.
`record`		The fields under `record` refer to the actual records for that Job ID. If you perform a Get Ingest Job Result operation after the Ingest Bulk operation, then the records display the record that was either inserted, upserted, updated, or deleted using bulk operation with the actual data that was provided by the user to Salesforce. This data is a part of the substitutable sObject in the records. If the Job ID is of a bulk query then it displays the records that were queried for the bulk query.
`islastBatch`	Boolean	This field if set to `true` indicates that there are no more batches to be retrieved and `false` if batches are remaining. The control group loops should be set on condition to check if this field has been updated to `true`.

Fault

On the **Fault** tab, you can find the error code and error message of the Salesforce Get Ingest Job Result activity. See Error Codes for more information about error codes and corrective actions to take.

Fault	Thrown when..
SalesforceRestException	There is a problem with REST request, response or any other REST API error.
SalesforceLoginException	An error occurred when logging in to the Salesforce.com server.
SalesforceBulkException	An error occurred when validating input parameters, requests, and parsing data before sending a request to Salesforce.

Salesforce Close Ingest Job Activity

The Salesforce Close Ingest Job activity is used to change the state of open jobs to a completed state. For example, UploadComplete, Closed, Aborted.

General

On the **General** tab, establish a connection with the Salesforce.com server.

The following table lists the configurations on the **General** tab of the Salesforce Close Ingest Job activity:

Field	Module Property ?	Description
Name	No	The name displayed as the label of the activity in the process.
Salesforce Connection	Yes	The path to the Salesforce shared resource. Click the **Choose/Create Default Resource** icon 🔍 to browse and select a connection.

Description

On the **Description** tab, enter a short description for the Salesforce Close Ingest Job activity.

Input

On the **Input** tab, specify the input values for the Salesforce Close Ingest Job activity.

The following table lists the input elements on the **Input** tab of the Salesforce Close Ingest Job activity:

Input Items	Data Type	Description
CloseJobConnection (All fields in this section are optional.)		
`ServerUrl`	String	Enter the web address of the endpoint that is used by this operation.
`SessionId`	String	Enter the unique ID associated with this session.
`ExternalSessionIdUsed`	Boolean	Specify whether an external session ID is used (`true`) or not (`false`). If the value is set to `true`, an external session ID is automatically filled in the **sessionId** field. 📝 If the ExternalSessionIdUsed is set to true, the session cannot be refreshed and any attempt to do so results in an exception.
CloseJob		
`jobId`	String	Enter the ID of the bulk job that is going to be UploadComplete, closed, or aborted.

Input Items	Data Type	Description
`state`	String	Enter the state to update the job to. **Valid values:** `UploadComplete`, `Closed`, or `Aborted` referring to either closing a job or aborting it.
`CloseJobOptional`		
`timeout`	Long	Specify the number of milliseconds an internal API call waits before returning the data. **Default value:** Infinite (that is, activity times out only after all records are processed)

Output

On the **Output** tab, you can find the output value.

The following table lists the output elements on the **Output** tab of the Salesforce Close Job activity:

Output Item		Data Type	Description
SalesforceCloseJobOutput			
jobInfo	`apiVersion`	String	The salesforce API version in which the job was created.
	`columnDelimiter`	String	Column Delimiter (symbol used to separate fields in the CSV records) used in the CSV job data. Values include COMMA, CARET, PIPE, BACKQUOTE, SEMICOLON, and TAB.
	`concurrencyMode`	String	For future use. Currently only parallel mode is supported in Bulk API 2.0.
	`contentType`	String	The format of the data being processed. Currently only CSV is supported for Bulk API 2.0.
	`contentUrl`	String	The URL to use for Upload Job requests for this job. Only provided if the job is in Open state.
	`createdById`	String	The ID of the user who created the job. The batch is created by using the same user.
	`createdDate`	DateTime	The date and time in the UTC time zone when the job was created.
	`externalIdFieldName`	String	Optional. This field contains the name of the field defined as the key indexing field in the Salesforce object. It is used with the upsert operation.

Output Item		Data Type	Description
	id	String	Unique ID of the job. The same can be mapped to further activities such as CheckStatus, GetResult, and CloseJob.
	jobType	String	The job's type. Values include: • BigObjectIngest: BigObjects job • Classic: Bulk API 1.0 job • V2Ingest: Bulk API 2.0 job
	lineEnding	String	The line ending used for CSV job data. This output field can be mapped to the GetResult activity.
	object	String	The Salesforce object type for the data being processed.
	operation	String	The processing operation specified. This can be mapped to further activities such as CheckStatus and GetResult.
	state	String	The current state of processing for the job. The BulkOperation and BulkQuery activities always update the job's state to UploadComplete. Other states are Open, Aborted, JobComplete, and Failed.
	systemModstamp	String	Date and time in the UTC time zone when the job finishes.

Fault

On the **Fault** tab, you can find the error code and error message of the Salesforce Close Ingest Job activity. See Error Codes for more information about error codes and corrective actions to take.

Fault	Thrown when..
SalesforceRestException	There is a problem with REST request, response or any other REST API error.
SalesforceLoginException	An error occurs when logging in to the Salesforce.com server.
SalesforceBulkException	An error occurred when validating input parameters, requests, and parsing data before sending a request to Salesforce.

Salesforce Composite Palette

The Salesforce Composite palette implements Salesforce Composite APIs. You can use the activities of this palette to improve the performance of your process by reducing the number of API calls.

Salesforce provides the following composite resources. They are described here in an increasing order of complexity and flexibility:

- Salesforce Composite Batch
- Salesforce Composite Tree
- Salesforce Composite Dependent

Before using the Salesforce Composite Palette, you must ensure that the Salesforce Connection shared resource is configured with valid credentials and latest metadata is fetched in the project. For more information, see the Salesforce Tools topic.

Salesforce Composite Batch

The Salesforce Composite Batch activity executes a set of independent subrequests in a single request. Information cannot be passed from one subrequest to another.

General

On the **General** tab, you must provide the following information to establish a connection with the Salesforce.com server:

Field	Module Property?	Description
Name	No	The name displayed as the label of the activity in the process
Salesforce Connection	Yes	The path to the Salesforce shared resource. Click **Choose/Create Default Resource** 🔍 to select a usable connection for the activity.

Description

On the **Description** tab, you can enter a short description for the Salesforce Composite Batch activity.

Input

On the **Input** tab, you can provide the following input values for the Salesforce Composite Batch activity:

Input Item	Data Type	Description
SalesforceSessionInput (All fields in this section are optional.)		

Input Item	Data Type	Description
`externalSessionUsed`	Boolean	Specifies whether an external session ID is used (`true`) or not (`false`).
		If the value is set to `true`, an external session ID is filled in the **sessionId** field.
		In this case, when the session has to be refreshed, the refreshed operation cannot be performed and an error occurs.
`sessionId`	String	The unique ID associated with this session
`serverURL`	String	The URL of the endpoint that is used by this operation
**jsonBody	batchRequests**		The body of the subrequest in JSON format or using Salesforce Composite Batch subrequest.
		For more information about creating batch requests, see Creating a Composite Dependent Subrequest
SalesforceOptionalInput (All fields in this section are optional.)		
`timeout`	Integer	The timeout value.
		It specifies the duration (in milliseconds) for which an internal API call waits before data is returned.
HttpHeader (All fields in this section are optional.)		Request headers and their values to include with the subrequest.
		For more information, see Salesforce Composite API documentation.
`key`	String	HTTP header name
`value`	String	HTTP header value

Creating a Salesforce Composite Batch Subrequest

1. To create a subrequest, in the upper right corner of the **Input** tab, click **Create Salesforce Composite Subrequest** .

2. In the Create Salesforce Composite Sub Request dialog box that is displayed, select the resource. Depending on the resource selected, provide further inputs and click **OK**.

- If the resource selected is queryAll or query then you can use the Query Builder to form the SOQL.

  For more information about using the Query Builder and its options, see Salesforce Query Builder.

- If the resource selected is search then you can use the Search Query Builder to form the SOSL.

  For more information about using the Search Query Builder and its options, see Salesforce Search Query Builder.

The following fields appear on the **Input** tab. For more information about these fields, see the Salesforce Composite API documentation.

Input Item	Data Type	Description
`batchRequests`		
`haltOnError`	Boolean	Specifies what to do when an error occurs while processing a subrequest
subRequest		
`url`	String	URL of the REST API endpoint
`method`	String	HTTP method (Examples: GET, POST)
`binaryPartName`	String	Name of the binary part in the multipart request
`binaryPartNameAlias`	String	Name parameter in the Content-Disposition header of the binary body part
`richInput`		Input for the request

Input Item	Data Type	Description
`jsonbody | binarybody? | sobject??`		Provide input for the subrequest body. **Condition:** The sobject is substituted with the object type only when you select the following items in the Salesforce Composite Subrequest wizard: • Resource: sobjects • Operation: Create or Update • Body Type: Complex

Output

The **Output** tab displays the activity results.

The following table lists the output elements on the **Output** tab of the Salesforce Composite Batch activity. For more information about these fields, see the Salesforce Composite API documentation.

Output Item	Data Type	Description
SalesforceCompositeBatchOutput		
`hasErrors`	Boolean	`true` - This value is displayed when at least one of the results in the result set is an HTTP status code in the range of `400` to `500`. `false` - This code is displayed when there are no errors.
results		Collection of subrequest results
`statusCode`	String	An HTTP status code for the current subrequest
`result`	String	Response body of this subrequest

Fault

On the **Fault** tab, you can find the error code and error message of the Salesforce Composite Batch activity. For more information about the error codes and the corrective actions to take, see Error Codes.

Fault	Thrown when..
SalesforceRestException	There is a problem with REST request, response or some other REST API issue.
SalesforceLoginException	An error occurs when logging in to the Salesforce.com server
SalesforceConnectionNotFoundException	An error occurs when the external session ID is not used or the Salesforce connection is not configured correctly.

Salesforce Composite Tree

The Salesforce Composite Tree activity creates one or more sObject trees with root records of the specified type.

General

On the **General** tab, you must provide the following information to establish a connection with the Salesforce.com server:

Field	Module Property?	Description
Name	No	The name displayed as the label of the activity in the process
Salesforce Connection	Yes	The path to the Salesforce shared resource. Click **Choose/Create Default Resource** 🔍 to select a usable connection for the activity.

Description

On the **Description** tab, you can enter a short description for the Salesforce Composite Tree activity.
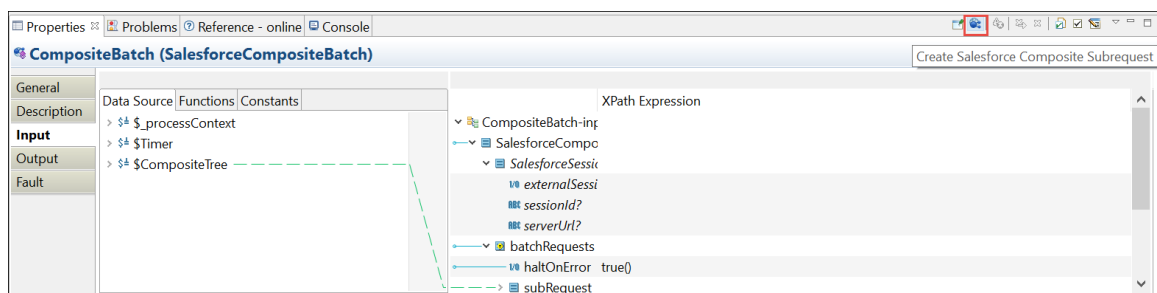
Input

On the **Input** tab you can specify the following input values for the Salesforce Composite Tree activity:

Input Items	Data Type	Description
SalesforceSessionInput (All fields in this section are optional)		
`externalSessionUrl`	String	The external web address of the endpoint that is used by this operation
`externalSessionId`	String	The external unique ID associated with this session

Input Items	Data Type	Description
`serverSessionUsed`	Boolean	Specifies whether an external session ID was used.
		If the value is set to `true`, an external ID is set in the **sessionId** field.
		In this case, when the session has to be refreshed, the refresh operation cannot be performed and an exception occurs.
SalesforceCompositeTree		
`sObjects`		
`rootObject`	Complex	Complex rootObject can be type substituted by a complex element from the `composite.xsd` schema, retrieved from Salesforce.com server.
		childRelationship: After the rootObject is type substituted, you can see the childRelationship element that contains the children of the substituted rootObject. Create sObject trees by type substituting the objects inside childRelationship.
SalesforceOptionalInput		
`timeout`	Integer	The timeout value.
		The duration (in milliseconds) for which an internal API call waits before the data is returned.
HttpHeader		Request headers and their values to include with the subrequest.
		For more information, see Salesforce API documentation.
`key`	String	HTTP header name
`value`	String	HTTP header value

Output

The **Output** tab displays the activity results.

The following table lists the output elements on the **Output** tab of the Salesforce Composite Tree activity. For more information on any of the fields, see the Salesforce Composite API documentation.

Output Item	Data Type	Description
SalesforceCompositeTreeOutput		
`hasErrors`	Boolean	`true` - This value is displayed when at least one of the results in the result set is an HTTP status code in the range of `400` to `500`.
		`false` - This code is displayed when there are no errors.

Output Item	Data Type	Description
results		Collection of results
referenceId	String	Reference ID of the requested record
id	String	ID of the requested record
errors		
fields	String	Fields for which there is an error
message	String	Error message
statusCode	String	If the request is not well-formed, the API returns a `400 Bad Request HTTP` status. Fix the syntax of the request and try again. If the request is well-formed, the API returns a `200 OK HTTP` status. If an item was processed successfully, the success flag for that item is displayed. Error information is returned in the error array.
extendedErrorDetails		More details about the error. Reserved for future use.
extendedErrorCode		Extended error code and extra error properties

Fault

On the **Fault** tab, you can find the error code and error message of the Salesforce Composite Tree activity. For more information about the error codes and the corrective actions to take, see Error Codes.

Fault	Thrown when..
SalesforceRest Exception	There is a problem with REST request, response or some other REST API issue.
SalesforceLogi nException	An error occurs when logging in to the Salesforce.com server.
SalesforceConn ectionNotFoun dException	An error occurs when the external session ID is not used or the Salesforce connection is not configured correctly.

Salesforce Composite Dependent

The Salesforce Composite Dependent activity runs a series of subrequests in a single call. You can use the output of one subrequest as an input to the next subrequest in the series.

General

On the **General** tab, you must provide the following information to establish a connection with the Salesforce.com server:

Field	Module Property ?	Description
Name	No	The name displayed as the label of the activity in the process
Salesforce Connection	Yes	The path to the Salesforce shared resource. Click **Choose/Create Default Resource** 🔍 to select a usable connection for the activity.

Description

On the **Description** tab, you can enter a short description for the Salesforce Composite Dependent activity.

Input

On the **Input** tab, you can specify the following input values for the Salesforce Composite Dependent activity:

Input Items	Data Type	Description
SalesforceSessionInput (All fields in this section are optional.)		
`externalSessionUsed`	Boolean	Specifies whether an external session ID is used. If the value is set to `true`, an external session ID is filled in the **sessionId** field. In this case, when the session has to be refreshed, the refreshed operation cannot be performed and an error occurs.
`sessionId`	String	The unique ID associated with this session
`serverUrl`	String	The web address of the endpoint that is used by this operation
compositeRequest		The body of the subrequest in JSON format or using Salesforce Composite Batch subrequest. For more information about creating dependent requests, see Creating a Composite Dependent Subrequest.
SalesforceOptionalInput (All fields in this section are optional.)		

Input Items	Data Type	Description
`timeout`	Boolean	The timeout value. It specifies the duration (in milliseconds) for which an internal API call waits before data is returned.
HttpHeader (All fields in this section are optional.)		Request headers and their values to include with the subrequest. For more information, see the Salesforce API documentation.
`key`	String	HTTP header name
`value`	String	HTTP header value

Creating a Salesforce Composite Dependent Subrequest

1. To create a subrequest, in the upper right corner of the **Input** tab, click **Create Salesforce Composite Subrequest** .

2. In the Create Salesforce Composite Sub Request dialog box that is displayed, enter the referenced ID and then select the resource. Depending on the resource selected, provide further inputs and click **OK**.
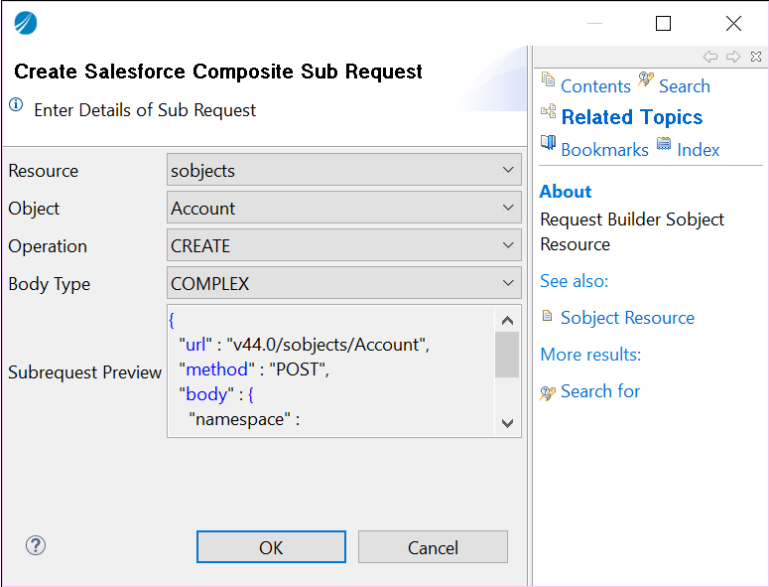
> If the resource selected is queryAll or query then you can use the Query Builder to form the SOQL.
>
> For more information about using the Query Builder and its options, see Salesforce Query Builder.

The following fields appear on the **Input** tab. For more information about these fields, see the Salesforce Composite API documentation.

Input Item	Data Type	Description
`compositeRequest`		
`allOrNone`	Boolean	Specifies what to do when an error occurs while processing a subrequest
`subRequest`		Subrequests are displayed in the order they are created
`referenceId`	String	Unique Reference ID that maps to the subrequest's response and can be used to refer to the response in later subrequests.
`url`	String	URL of the REST API endpoint
`method`	String	HTTP method (Examples: GET, POST)
`httpHeaders`		Request headers and their values to include with the subrequest
`key`	String	HTTP header name

Input Item	Data Type	Description
`value`	String	HTTP header value
`body`		
`sobject`		Provide input for the subrequest body. **Condition:** The sobject is substituted with the object type only when you select the following items in the Salesforce Composite Subrequest wizard: • Resource: sobjects • Operation: Create or Update • Body Type: Complex

Output

The **Output** tab displays the activity results.

The following table lists the output elements on the **Output** tab of the Salesforce Composite Dependent activity:

Output Item	Data Type	Description
SalesforceCompositeOutput		
compositeResponse		
`referenceId`	String	The ID that maps to the subrequest's response
`httpStatusCode`	String	An HTTP status code for the current subrequest
httpHeaders		Request headers and their values to include with the subrequest

Output Item	Data Type	Description
`key`	String	HTTP header name
`value`	String	HTTP header value
`body`	String	Response body of the subrequest in JSON format

Fault

On the **Fault** tab, you can find the error code and error message of the Salesforce Composite Dependent activity. For more information about the error codes and the corrective actions to take, see Error Codes.

Fault	Thrown when..
SalesforceRest Exception	There is a problem with REST request, response or some other REST API issue.
SalesforceLogi nException	An error occurs when logging in to the Salesforce.com server.
SalesforceConn ectionNotFoun dException	An error occurs when the external session ID is not used or the Salesforce connection is not configured correctly.

Samples

The ActiveMatrix BusinessWorks Plug-in for Salesforce.com installer includes some sample projects. The samples help you understand how this plug-in operates.

To run the following sample projects smoothly, apply for a new developer edition account of Salesforce.com.

The following sample projects are located in the `TIBCO_HOME`\bw\palettes\salesforce\`version_number`\samples directory:

- Topic Subscriber Project
- BulkSample Project
- Bulk2.0Sample Project
- GeneralSample Project
- RelationshipQuery Project
- ERPIntegration Project
- WorkWithInvokerActivity Project
- SalesforceCompositeSample Project

Working with TopicSubscriberSample Project

The TopicSubscriberSample project gives you a quick overview of how to use the Salesforce Subscriber activity of the TIBCO ActiveMatrix BusinessWorks™ Plug-in for Salesforce.com.

Procedure

1. Start TIBCO Business Studio.
2. Click **File** > **Import**.
3. In the Import window, expand the **General** folder and select the **Existing Studio Projects into Workspace** item. Click **Next**.
4. Click **Browse** next to the **Select archive file** field to locate the samples. Click **Finish**.

   The sample project is located in the `TIBCO_HOME`\bw\palettes\salesforce\`version_number`\samples\TopicSubscriberSample directory.

5. Change the values in the **User Name** and **Password** fields in the Connection Configuration panel of the Salesforce Connection shared resource.
6. In this project, five processes are created under three packages, `initpushtopic`, `topicsubscribersample` and `filtertopicsubscribersample`. Configure these processes as explained below:

   - CreatePushTopic Process

     This process creates a push topic on the Salesforce server with the name provided.

   - AccountUpdates Process

     This process creates a record, updates it, and then deletes the record in the end.

   - SubscribeToAccountUpdates Process
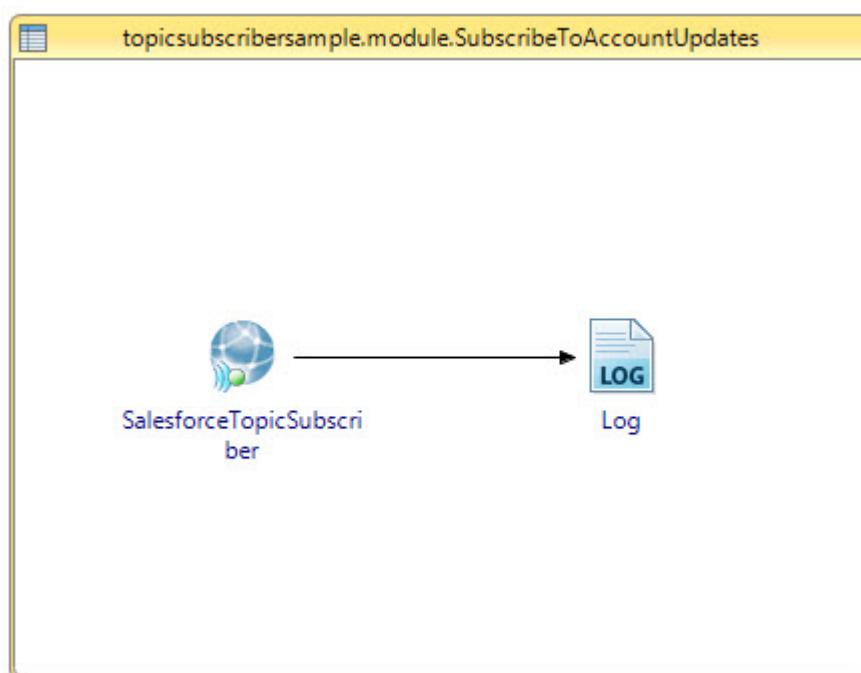
This process provides the same topic name that was created in CreatePushTopic process. It subscribes to the topic name and generates a process event for each message received from AccountUpdates process. It starts receiving messages for all updates to the object in the last 24 hours.

- AccountUpdatesForFilter Process

  This process creates a record with the filter string set to name. Then it first creates a record that does not have a filter string and then updates it to contain a filter string. Both the records are deleted at the end.

- FilteredTopicSubscriber Process

  In this process, the topic name AccountUpdates is provided. A filter string is also provided to validate the condition. Note that the topic name is the same as that provided in the CreatePushTopic process. It subscribes to the topic name and generates a process event for each filtered message received from AccountUpdatesForFilter process. It starts receiving messages for all updates to the object in the last 24 hours.

7. Test and run the processes in the project, in the order they have been listed earlier.

Working with BulkSample Project

The BulkSample project gives you a quick overview of how to use the Salesforce Bulk API Palette of TIBCO ActiveMatrix BusinessWorks Plug-in for Salesforce.com.

Procedure

1. Start TIBCO Business Studio and go to **File** > **Import**.

2. In the Import window, expand the **General** folder and select the **Existing Studio Projects into Workspace** item and click **Next**.

3. Click **Browse** next to the **Select archive file** field to locate the samples. Click **Finish**.
    The sample project is located in the `TIBCO_HOME\bw\palettes\salesforce\version_number \samples\BulkSample` directory.

4. Change the values in the **User Name** and **Password** fields in the Connection Configuration panel of the Salesforce Connection shared resource.

5. Configure the processes that are created:

    1. BulkInsertJson Process

        This process inserts records for the contact sObject using JSON format and queries the record using Bulk Query.

        1. Provide the `IBCO_HOME\bw\palettes\salesforce\`*`version_number`*`\samples\BulkSample\sampleDataJson.json` file in the BulkInsert activity.

        2. Create a file `test.json` on your system.

        3. Provide that location in the Output File of Insertgetresult activity.
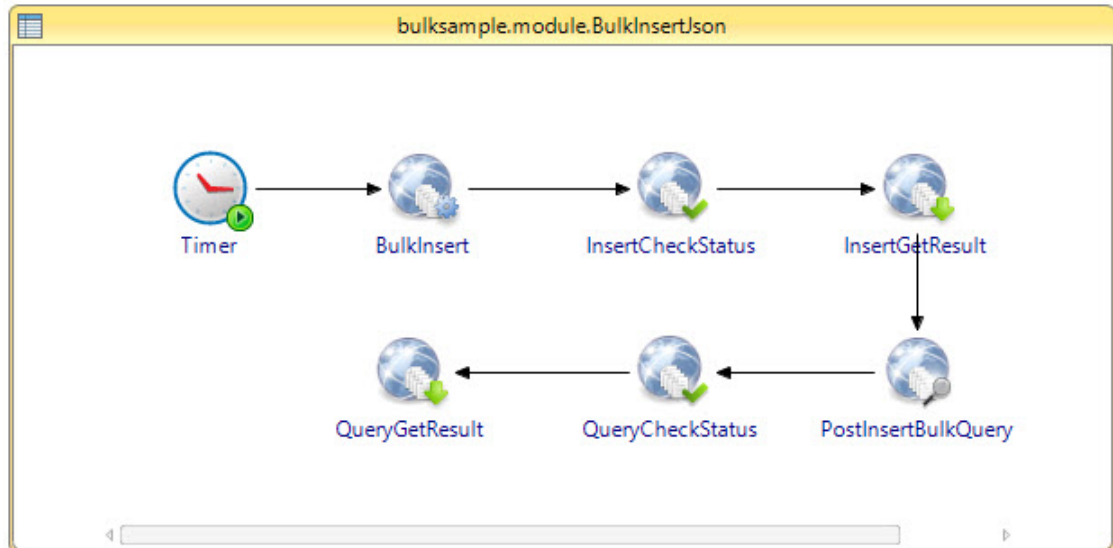
    2. BulkUpdateRow Process

        This process updates the record inserted by the above process by using the Update operation of the Bulk Operation activity with ROW format.

    3. BulkUpsertRow Process

        This process upserts the record inserted by the above process by using the Upsert operation of the Bulk Operation activity with ROW format. You should create a file test.xml on your system and provide that location in the Output File on the **General** tab of Upsertgetresult activity.

    4. BulkUpsertXML Process

        This process updates the inserted records and inserts a new record by using the Upsert operation of the Bulk Operation activity. You should create a file `sampledata.xml` on your system and provide that location in the Output File of Upsertgetresult activity.

    5. BulkDeleteCSV Process

        Using the Bulk Operation activity's Delete operation, this process deletes the records that were inserted in the BulkInsertJson and BulkUpsertXML processes. You should create a file deletedata.csv on your system and provide that location in the Output File on the General tab of Upsertgetresult activity.

    6. BulkGetSubsetResult Process

        The process first creates records named from TibcoBW1 to TibcoBW11. Then a bulk query is run on these records with chunkSize set to 4. The result for this query is processed in subsets of 3 (meaning that each activity loop processes 3 batches) until all batches are processed. The results are logged, and the records are deleted as part of cleanup. After the records are processed, the job created by Salesforce Bulk Query is closed by using the Salesforce Close Job activity.

6. Test and run the processes in the project, in the order they are listed above.

Working with Bulk2Sample Project

The Bulk2Sample project gives you a quick overview of how to use the Salesforce Bulk API 2.0 Palette of ActiveMatrix BusinessWorks Plug-in for Salesforce.com.

Procedure

1. Start TIBCO Business Studio and go to **File** > **Import**.
2. In the Import window, expand the **General** folder and select the **Existing Studio Projects into Workspace** item and click **Next**.
3. Click **Browse** next to the **Select archive file** field to locate the samples. Click **Finish**.
   The sample project is located in the `TIBCO_HOME\bw\palettes\salesforce\version_number \samples\BulkSample` directory.
4. Change the values in the **User Name** and **Password** fields in the Connection Configuration panel of the Salesforce Connection shared resource.
5. Edit the **bulkInputFile** module property to point to the `TIBCO_HOME\bw\palettes\salesforce \version_number\samples\Bulk2_0Sample\bulkInsert.csv` file provided in the Bulk2_0Sample. Modify the **outputFilePath** module property to point to any empty csv file in the system. Save the project.
6. Configure the processes that are created:

   1. BulkOperationCSV Process
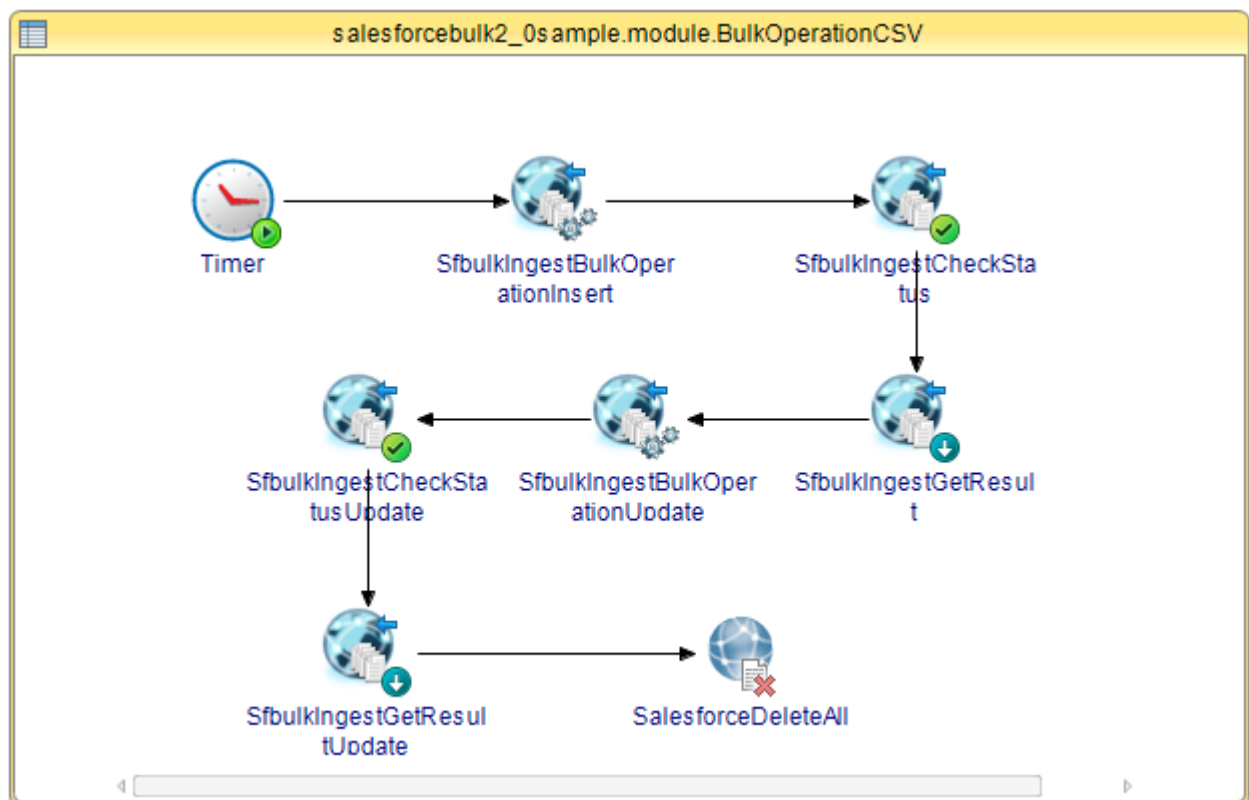
   This process creates an Account object named BulkInsertAccount by using the `bulkInsert.csv` file provided in the module property. It then updates the account object's name to Updated Account by using the sObject mapped in the input section of SfBulkIngestBulkOperationUpdate activity. The

result of the update activity is retrieved and the created and updated object is deleted as part of cleanup.

2. SubscriberTrigger Process

This process creates an account named AccountNew to trigger the BulkQuery process starter Subscriber.

3. BulkQuery Process

Run this process after running SubscriberTrigger at least once. The Subscriber is triggered when an account with the name AccountNew is created. The Bulk 2.0 query queries for all accounts with the name AccountNew. The results are fetched with Get Result and printed by using the Log activity.

4. SfbulkGetBatchedResult Process

In this process 20 account objects with the name BatchedAccount are inserted. Then queried by using Bulk 2.0 Query and the results are retrieved in batches using Bulk 2.0 Get Result with *maxRecords* fetched in one iteration limited to 5. The results are written to the file provided in the outputFilePath module property.

5. Cleanup Process

Run this process after running the BulkOperationCSV, SubscriberTrigger, BulkQuery, and SfbulkGetBatchedResult processes. This cleans up the data created in these processes.

7. Test and run the processes in the project, in the order they are listed above.

Working with GeneralSample Project

This GeneralSample project gives you a quick overview of how to use TIBCO ActiveMatrix BusinessWorks Plug-in for Salesforce.com.

Procedure

1. Start TIBCO Business Studio.

2. Click **File** > **Import**.

3. In the Import window, expand the **General** folder and select the **Existing Studio Projects into Workspace** item. Click **Next**.

4. Click **Browse** next to the **Select archive file** field to locate the samples. Click **Finish**.

   The sample project is located in the `TIBCO_HOME\bw\palettes\salesforce\version_number \samples\GeneralSample` directory.

   In this project, two processes are created:

   * General Process

     This process is used to show the general configurations of Salesforce activities. Most Salesforce activities are included, such as Salesforce Create All, Salesforce Query All, Salesforce Retrieve All, and so on.

   * RemoveSampleData Process

     After running the General process, you can run this process to remove all generated data from your Salesforce.com database.

5. Change the values in the **User Name** and **Password** fields in the Connection Configuration panel of the Salesforce Connection shared resource.

6. Test and run the processes in the project.

General Process

This General process is used to show the general configurations of Salesforce activities.

The process is designed with the following activities:
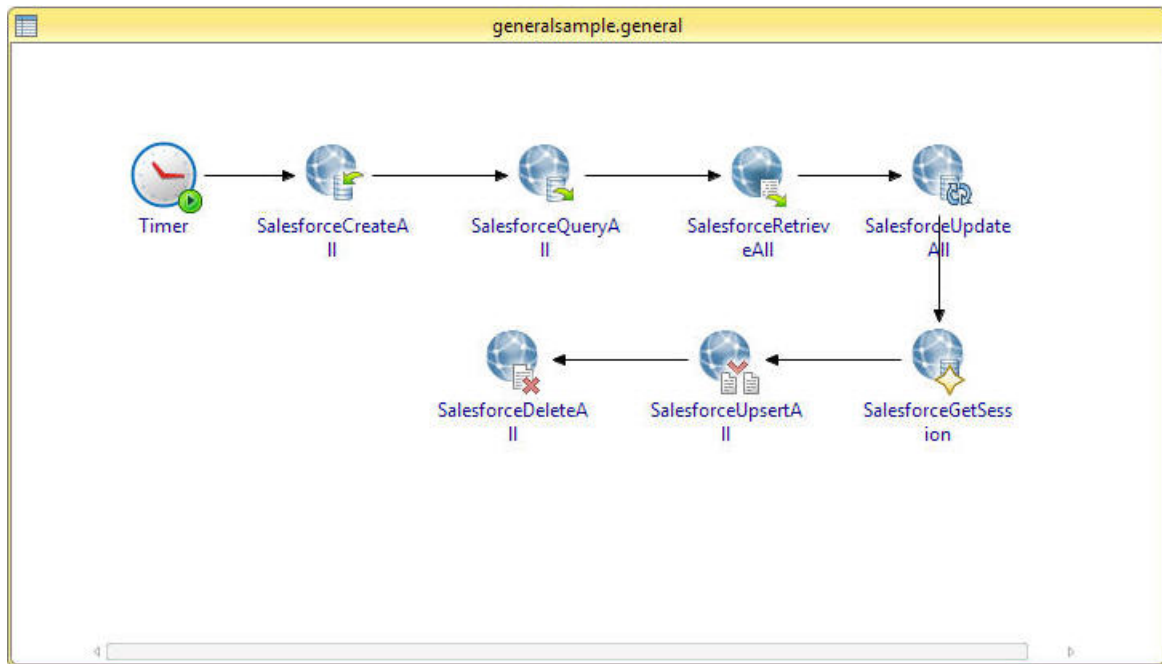
The following table lists and explains the activities in the process.

Activity	Description
Salesforce Create All	Salesforce Create All activity. Creates five sample contact sObjects in your Salesforce.com database.
Salesforce Query All	Salesforce Query All activity. Executes a query to get the five sample contact sObjects created by the Salesforce Create All activity.
Salesforce Retrieve All	Salesforce Retrieve All activity. Retrieves the five sample contact sObjects created by the Salesforce Create All activity.
Salesforce Update All	Salesforce Update All activity. Updates the information of the five sample contact sObjects.
Salesforce Get Session	Salesforce Get Session activity. Gets a session from the session pool that is related to the specified Salesforce connection. This session is used in the following Salesforce Upsert All activity.

Activity	Description
Salesforce Upsert All	Salesforce Upsert All activity.
	Upserts the information of the five sample contact sObjects.
	When doing this activity, since ID is defined as the external ID, the five sample contact sObjects are updated.
Salesforce Delete All	Salesforce Delete All activity.
	Deletes all sample data from your Salesforce database.

Working with RelationshipQuery Project

The RelationshipQuery project shows how to do relationship queries in TIBCO ActiveMatrix BusinessWorks Plug-in for Salesforce.com. Two kinds of relationship queries are defined by Salesforce.com: parent-to-children and child-to-parent.

Procedure

1. Start TIBCO Business Studio.
2. Click **File** > **Import**.
3. In the Import window, expand the **General** folder and select the **Existing Studio Projects into Workspace** item. Click **Next**.
4. Click **Browse** next to the **Select archive file** field to locate the samples. Click **Finish**.

   The sample project is located in the `TIBCO_HOME`\bw\palettes\salesforce\`version_number` \samples\RelationshipQuerySample directory.

   In this project, the following four processes are created:

   - CreateData Process

     Before running the childToParent process or the parentToChildren process, run this process to create sample sObjects with the relationship in your Salesforce.com database.

   - RemoveData Process

     After running the childToParent process or the parentToChildren process, you can run this process to remove all the sample data from your Salesforce.com database.

   - ChildToParent Process

     This process shows how to do a child-to-parent relationship query and how to map the data to other activities.

     See Relationship Query for more details.

   - ParentToChildren Process

     This process shows how to do a parent-to-children relationship query and how to map the data to other activities.

     See Relationship Query for more details.

5. Change the values in the **User Name** and **Password** fields in the Connection Configuration panel of the Salesforce Connection shared resource.
6. Test and run the processes in the project.

> If some fields cannot be accessed when running the project, you might have to change the Field Accessibility configuration in the Salesforce.com database.

ChildToParent Process

This ChildToParent process shows how to do a child-to-parent relationship query and how to map the data to other activities.

The process is designed with the following activities:

The following table lists and explains the fields in the activities in the process.

Activity	Description
Salesforce Query All	Salesforce Query All activity.
	Processes the child-to-parent relationship query.
	The following statement is entered in the `queryString` field.
	`Select Id, FirstName, LastName, Account.Id, Account.Name, Account.BillingStreet from Contact where Account.Name='TIB_Sample_TIB' and Account.BillingStreet='TIB_Sample_TIB'`
Mapper	Receives the returned data from the Salesforce Query activity, and then maps the data values.
	For detailed information, see Mapping Data in the ChildToParent Process.

Mapping Data in the ChildToParent Process

Mapper is a general activity defined in TIBCO ActiveMatrix BusinessWorks. It is used as the data consumer of the Salesforce Query All activity.

To map data values in the ChildToParent process, complete the following steps:

Procedure

1. Click the Mapper activity in the ChildToParent process, the Mapper panel is displayed.

   On the **Input** tab, you can see the available data source and the expected input of the activity. The data source and activity input are represented as schema trees.

2. Cast the **records** node that is received from the Salesforce Query All activity into the data type of the Contact sObject.

3. Map the cast **records** element to the **Contacts** element in the activity input area.

4. Map other elements to the corresponding elements in the activity input area, as shown in the following figure.

Result

Map data in the ChildToParent process:

ParentToChildren Process

This ParentToChildren process shows how to do a parent-to-children relationship query and how to map the data to other activities.

The process is designed with the following activities:

The following table lists and explains the fields in the activities in the process.

Activity	Description
Salesforce Query All	Salesforce Query All activity.
	Processes the parent-to-child relationship query.
	The following statement is entered in the `queryString` field.
	`Select Id, Name,BillingStreet, (select Id, FirstName,LastName from Contacts) from Account a where a.Name='TIB_Sample_TIB' and a.BillingStreet='TIB_Sample_TIB'`
Mapper	Receives the returned data from the Salesforce Query All activity, and then maps the data values.
	For detailed information, see Mapping Data in the ParentToChildren Process.

Mapping Data in the ParentToChildren Process

Mapper is a general activity defined in TIBCO ActiveMatrix BusinessWorks. It is used as the data consumer of the Salesforce Query activity.

To map data values in the ParentToChildren process, complete the following steps:

Procedure

1. Click the Mapper activity in the ParentToChildren process, the Mapper panel is displayed.

   On the **Input** tab, you can see the available data source and the expected input of the activity. The data source and activity input are represented as schema trees.

2. Cast the **records** element that is received from the Salesforce Query activity into the data type of the Account sObject and then map the cast **records** element to the **Account** element.

3. Map other elements to the corresponding elements.

   If you want to map a complex element under the cast element, you have to manually set XPath in the corresponding input field, as shown in the following figure. The XPath must be matched with the corresponding schema returned from the query process at run time.
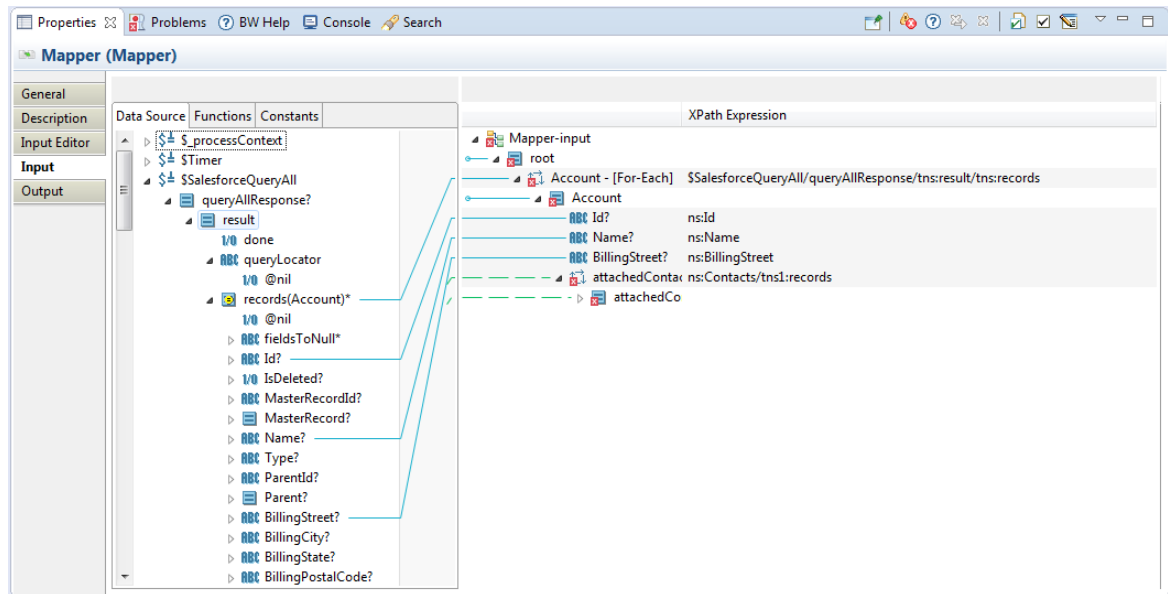
Working with ERPIntegration Project

The ERPIntegration project shows how to use a TIBCO ActiveMatrix BusinessWorks™ process to manipulate the ActiveMatrix BusinessWorks Plug-in for Salesforce.com service.

To access the sample application, complete the following steps:

Procedure

1. Start TIBCO Business Studio.

2. Click **File** > **Import**.

3. In the Import window, expand the **General** folder and select the **Existing Studio Projects into Workspace** item. Click **Next**.

4. Click **Browse** next to the **Select archive file** field to locate the samples. Click **Finish**.

   The sample project is located in the `TIBCO_HOME\bw\palettes\salesforce\version_number\samples\ERPIntegration` directory.

5. Log on the Salesforce.com website using the new account information. Make sure that at least one product has been added to the price book, and this product has to be added as the line item for an Opportunity whose stage is `Close Won`.

6. Add a customized field named **ErpAccountId** in the Account Object on the Salesforce.com server.

7. To run the SyncOpportunity_OutboundListener process in TIBCO Business Studio, create an Outbound Message on the Salesforce.com website.

   Make sure that the Endpoint URI is the same as the one defined in the WSDL file of the Outbound Listener activity. For example, the Endpoint URI defined in the WSDL file is: `/OpportunityToOrder_UsingOutboundMsgListener/SyncOpportunityProcess_UsingOutboundMsgListener`, so in the Salesoforce.com website, the whole

URL address is `http://host:port/OpportunityToOrder_UsingOutboundMsgListener/SyncOpportunityProcess_UsingOutboundMsgListener`.

8. Generate an Outbound Message WSDL and configure the workflow rule to trigger the Outbound Message you just created.

> When you generate an Outbound Message WSDL, ensure that the following six fields are not added to the **Selected Fields** section:
>
> • **HasOpenActivity**
> • **HasOverdueTask**
> • **IsExcludedFromTerritory2Filter**
> • **LastReferencedDate**
> • **LastViewedDate**
> • **Territory2Id**

9. Use the `CreateErpObjects.sql` file to build Oracle database tables and other necessary objects. `CreateErpObject.sql` is available from the `TIBCO_HOME\bw\palettes\salesforce\version_number\samples\ERPIntegration` directory.

> For running this sample application, a new user of Salesforce.com creates at least one product as the line item for an Opportunity with the state of `Close Won`.

10. Change module properties if necessary in TIBCO Business Studio.

   • **BW_LastPollingTime_File**: the location of the file, in which the last polling time is recorded.
   • **BW_OpportunityStart_Time**: the time that the opportunities start to record.

   All opportunities occur before this time is ignored.

11. Configure Salesforce Connection, JDBC Connection and HTTP Connector.

12. Configure the JDBC driver for the JDBC activities both at design time and run time.
   See *TIBCO ActiveMatrix BusinessWorks Bindings and Palettes Reference* for more details.

13. Test and run the processes in the sample application.

Customer Master

The company's sales department uses Salesforce.com CRM to manage the sales process and to monitor the pipeline. Sales users create new Account records to represent prospects and Opportunity records to track deals in progress. The Account data entered by sales users is not accurate enough to satisfy the requirements for the back office fulfillment and finance departments, whereas the Oracle ERP data is tightly controlled and well maintained.

The Customer Master Integration requires implementation of two unidirectional processes on the TIBCO ActiveMatrix BusinessWorks platform.

   • SyncAccountProcess
   • UpdateAccountProcess

SyncAccountProcess Process

The SyncAccountProcess process copies all Accounts ready for promotion from the Salesforce.com server to the ERP system, writes back the ERP Account ID primary key field to the ErpAccountId field that has been created in the Salesforce.com Account, and changes the Account record type in the Salesforce.com server to the read-only Integrated Account type. This process is invoked from within the Opportunity to Order process as needed.

The process is designed with the following activities:
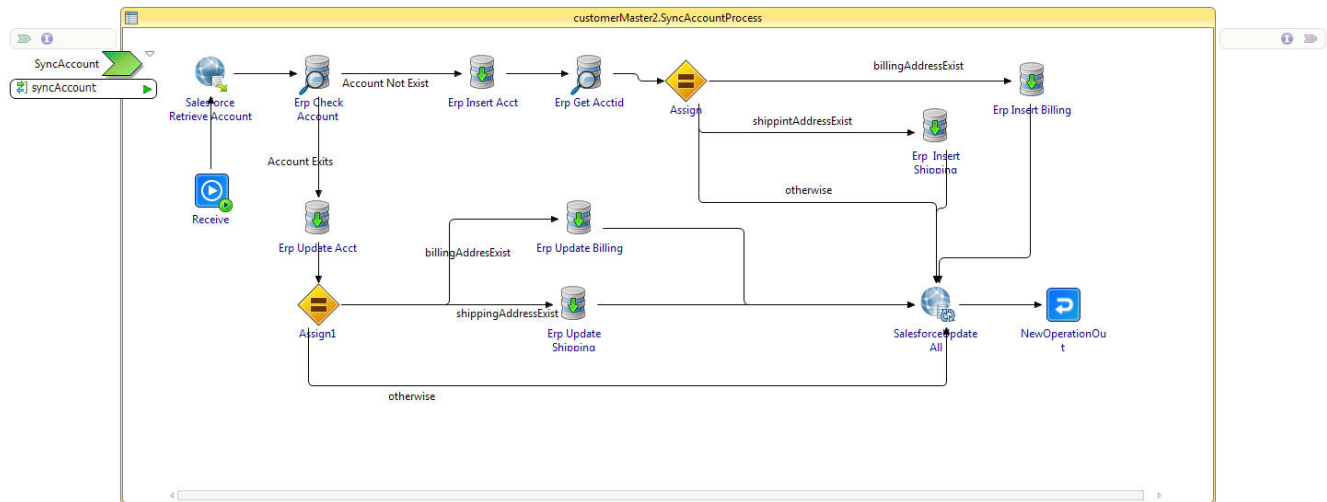
The following table lists and explains the activities in the process.

Activity	Description
Salesforce Retrieve Account	Salesforce Retrieve All activity. Gets a set of Salesforce.com Account sObject by using the given sObject IDs.
Erp Check Account	Checks whether an Account in the ERP_ACCT table with the same name as the getting Account from the Salesforce.com server exists.
Erp Update Acct	Updates the ERP_ACCT table while an Account is found in the activity named **Erp Check Account**.
Erp Update Shipping	Updates the shipping information in the ERP_ADDR table after executing the activity named **Erp Update Acct**.
Erp Update Billing	Updates the billing information in the ERP_ADDR table after executing the activity named **Erp Update Acct**.
Erp Insert Acct	Creates an Account in the ERP_ACCT table if an account is not found in the activity named **Erp Check Account**.
Erp Get AcctId	Gets the Account ID from the ERP_ACCT table after executing the activity named **Erp Insert Acct**.
Erp Insert Shipping	Creates the shipping information in the ERP_ADDR table when needed.
Erp Insert Billing	Creates the billing information in the ERP_ADDR table when needed.
Salesforce Update All	Salesforce Update All activity. Updates the ACCT_ID field value of the newly created records in the ERP_ACCT table to the ErpAccountId field in the Salesforce.com Account.

UpdateAccountProcess Process

The UpdateAccountProcess process monitors Accounts in the ERP system for changes and updates the Salesforce.com Account accordingly.

The process is designed with the following activities:
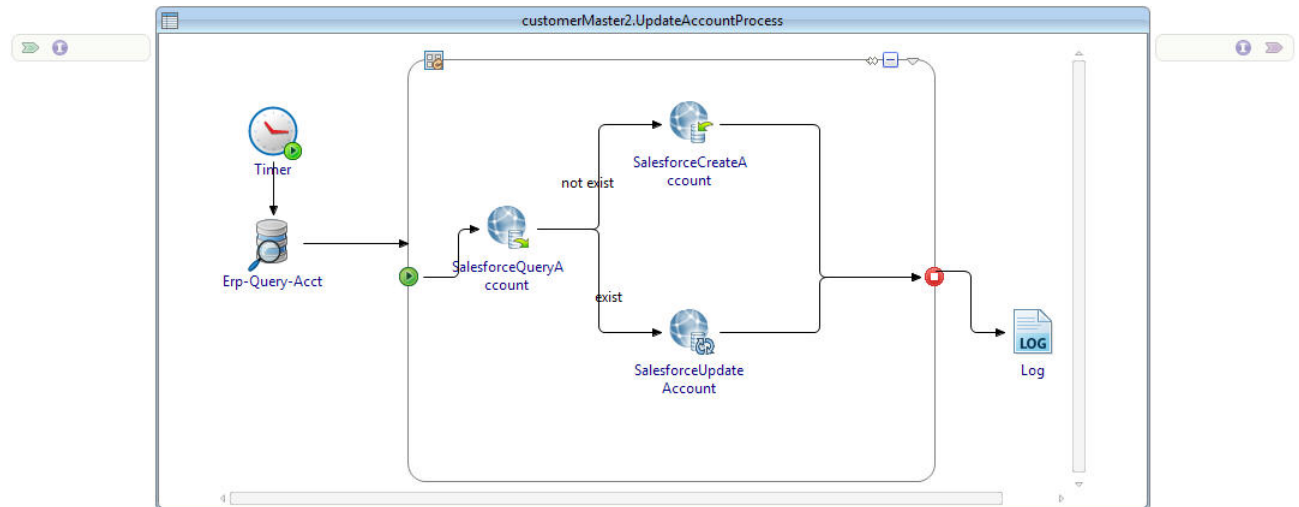
The following table lists and explains the activities in the process.

Activity	Description
Erp-Query-Acct	Gets ERP Accounts from the ERP_ACCT and ERP_ADDR tables.
Group	Repeats activities in the group.
Salesforce Query Account	Checks whether the ERP Account exists in the Salesforce.com Account.
Salesforce Create Account	If the ERP Account does not exist in the Salesforce.com server, this activity creates a new Account in it.
Salesforce Update Account	If the ERP Account exists in Salesforce.com, this activity updates the Account in it.

Opportunity to Order

During the selling process, a salesperson tracks the status of a deal as it moves through the sales pipeline using the Salesforce.com Opportunity object. When a deal closes, a new Order must be created in the ERP system to initiate the invoicing and shipment processes. The current manual entry process can be replaced by an automated data transfer from the Salesforce.com into the ERP system.

It concludes the following processes:

- OpportunityToOrder_Polling
- OpportunityToOrder_UsingOutbound

OpportunityToOrder_Polling

When a deal closes successfully, the owner of the Salesforce.com Opportunity representing that deal changes its status to `Closed Won`. The Opportunity to Order process requires a process to be implemented on the Business works platform that looks for newly closed opportunities on a regular basis and create a

new order in the ERP system each time one is found. If the Order's Account is not in the ERP system, the Account record has to be migrated from the Salesforce.com CRM as well, according to the Customer Master Process.

The process named SyncOpportunity_Polling takes this task in hand. Another two processes named GetLastPollingTime and SetLastPollingTime are called by the SyncOpportunity_Polling process.

GetLastPollingTime Process

The GetLastPollingTime process is called by the SyncOpportunity_Polling process.

The process is designed with the following activities:
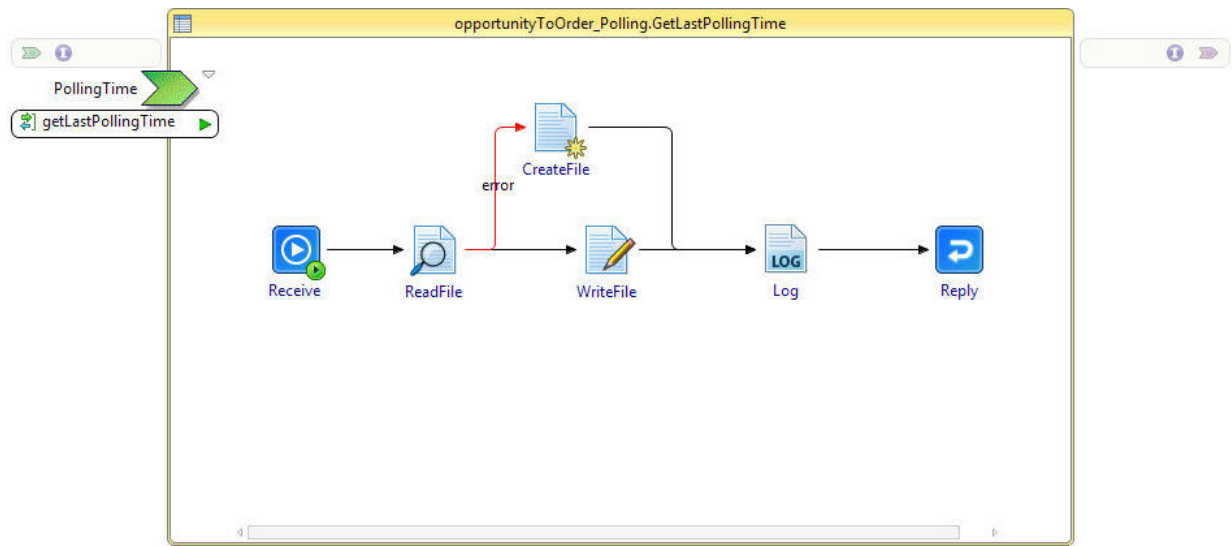
The following table lists and explains the activities in the process.

Activity	Description
Read File	Reads the file, in which the polling time is recorded.
Create File	Creates a file to record the polling time if it does not exist before.
Write File	Adds a lock sign to the file, in which the polling time is recorded.

SetLastPollingTime Process

The SetLastPollingTime process is called by the SyncOpportunity_Polling process.

The process is designed with the following activities:
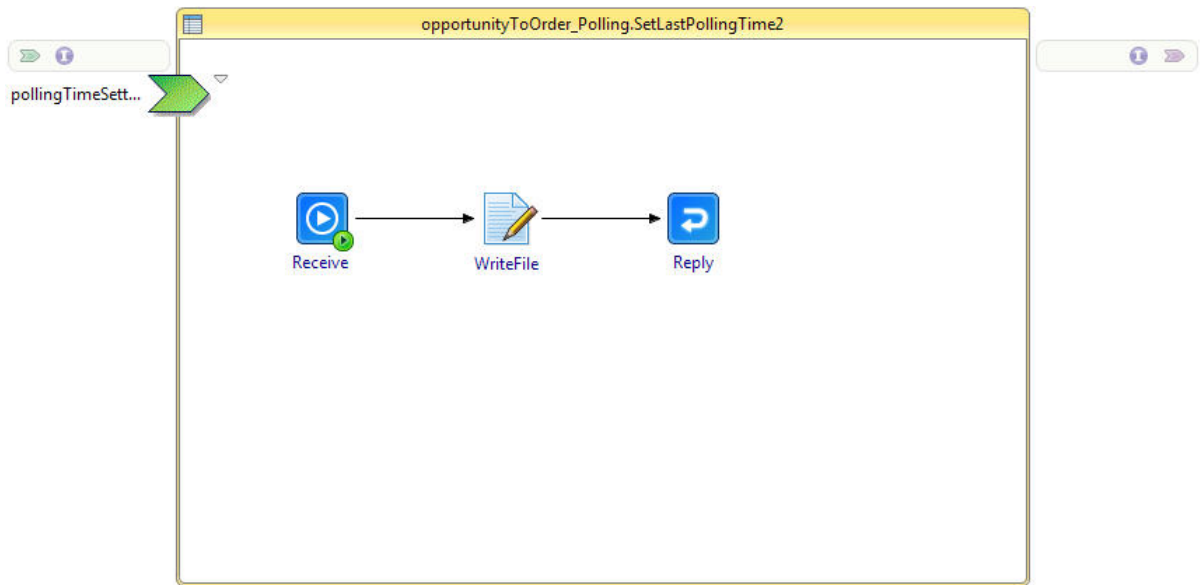
The following table lists and explains the activity in the process.

Activity	Description
Write File	Removes the lock sign from the file.
	Writes the current polling time in the file.

SyncOpportunity_Polling Process

The SyncOpportunity_Polling process takes the previously mentioned task in hand.

The process is designed with the following activities:
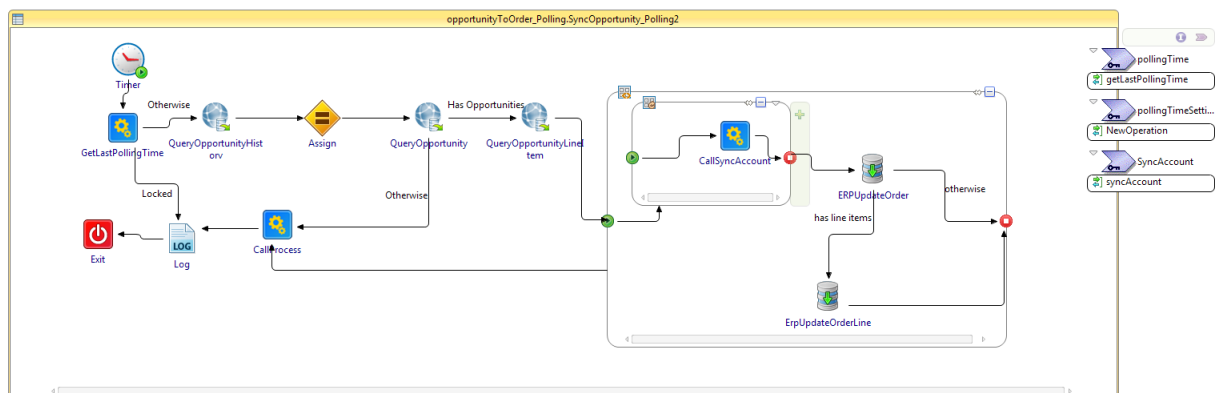
The following table lists and explains the activities in the process.

Activity	Description
GetLastPollingTime	Gets the last polling time and locks this process.

Activity	Description
QueryOpportunityHistory	Salesforce Query All activity.
	Gets the results of the opportunity history when the following conditions are met:
	`OpportunityHistory.State = 'Close Won'` `OpportunityHistory.CreateDate > the last polling time`
QueryOpportunity	Salesforce Query All activity.

	Gets results of the opportunities according to the following conditions:
	`Opportunity.Id in OpportunityHistory(Queried`
	`before).OpportunityId Opportunity.HasLineitem = true`

QueryOpportunityLineItem	Salesforce Query All activity.
	Gets results of the opportunity line items.
Transition Control	Group
ErpUpdateOrder	Updates the ERP_ORDER table in the ERP system.
ErpUpdateOrderLine	Updates the ERP_ORDERLINE table in the ERP system.

OpportunityToOrder_UsingOutbound

This OpportunityToOrder_UsingOutbound process uses the outbound message listener to complete the same task as the task described in the OpportunityToOrder_Polling section.

The process named SyncOpportunity_OutboundListener takes the task in hand.

SyncOpportunity_OutboundListener Process

The SyncOpportunity_OutboundListener process takes the task in hand.

The process is designed with the following activities:
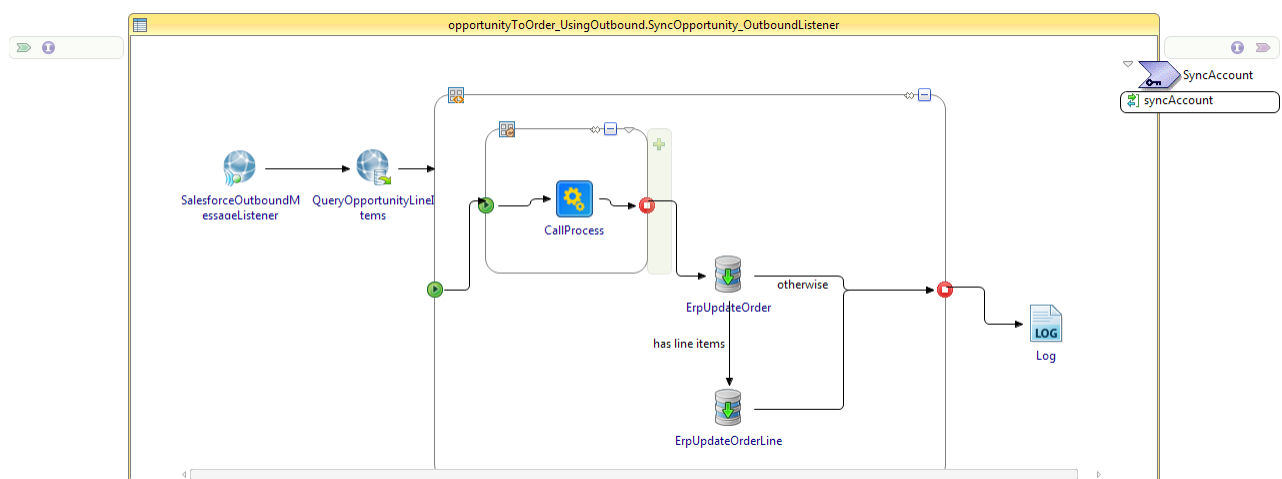
Activity	Description
Salesforce Outbound Message Listener	Sets an outbound message listener. You can use it to create a new job for processing the coming outbound message.
QueryOpportunityLineItems	Salesforce Query All activity. Gets results of the opportunity line items.
Transition Control	Group
ErpUpdateOrder	Updates the ERP_ORDER table in the ERP system.
ErpUpdateOrderLine	Updates the ERP_ORDERLINE table in the ERP system.

Working with WorkWithInvokerActivity Project

This WorkWithInvokerActivity project shows how to work with a SOAP activity in TIBCO ActiveMatrix BusinessWorks process to manipulate the TIBCO ActiveMatrix BusinessWorks Plug-in for Salesforce.com service.

To access the sample application, complete the following steps:

Procedure

1. Start TIBCO Business Studio.
2. Click **File** > **Import**.
3. In the Import window, expand the **General** folder and select the **Existing Studio Projects into Workspace** item. Click **Next**.
4. Click **Browse** next to the **Select archive file** field to locate the samples. Click **Finish**.

   The sample project is located in the `TIBCO_HOME\bw\palettes\salesforce\version_number \samples\WorkWithInvokerActivity` directory.
5. Log on the Salesforce.com website using the new account information. Ensure that at least one Contact record with the phone number (`4159017000`) used in the example process has been created.
6. Change the values in the **User Name** and **Password** fields in the Connection Configuration panel of the Salesforce Connection shared resource.
7. Test and run the processes in the sample application.

Workwithbwinvoker Process

The Workwithbwinvoker process in the project updates the phone number on the specified Contact records. It searches the Contact records with a specified phone number from the Salesforce.com server by using the Invoke activity, and then updates those Contact records with the new phone number.

The process is designed with the following activities:
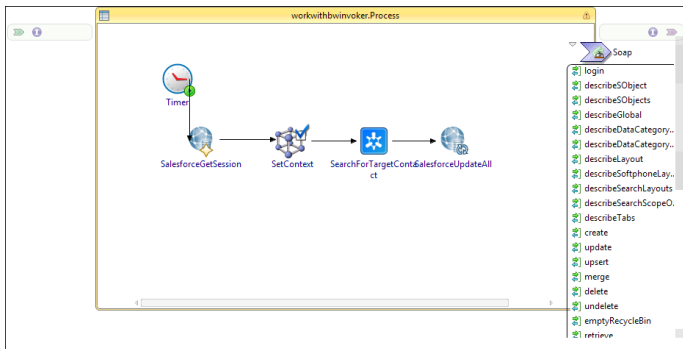
The following table lists and explains the activities in the process.

Activity	Description
SalesforceGetSession	Gets an existing session from the session pool that is related to the specified Salesforce connection.
SetContext	Sets the context parameters for the Invoke activity.
SearchForTargetContact	Searches the Contract records with the specified searchString. This search operation has been defined in the Partner WSDL used in this sample project.
SalesforceUpdateAll	Salesforce Update All activity. Updates the specified Contact records with the new phone number.

Working with the SalesforceCompositeSample Project

The SalesforceCompositeSample project shows how to use the Salesforce Composite API palette of ActiveMatrix BusinessWorks™ Plug-in for Salesforce.com. There are two processes in this project. They are independent of each other and can be run in any sequence.

To access the sample application, perform the following steps:

Procedure

1. In TIBCO Business Studio for BusinessWorks for BusinessWorks, click, **File** > **Import**
2. In the Import wizard, perform the following steps:
   a) Expand the **General** folder and select **Existing Studio Projects into Workspace** and click **Next**.
   b) Click **Browse** next to the **Select archive file** field to locate the samples and click **Finish**.

   > The sample project is located in the *TIBCO_HOME*\bw\palettes\salesforce*version_number*\samples\SalesforceCompositeSample directory.

   c) In the **Connection Configuration** panel of the Salesforce Connection shared resource, change the values in the **User Name** and **Password** fields.
   d) Test and run the two processes created in the project.

AllCompositeActivitiesDemo Process

The AllCompositeActivitiesDemo process demonstrates how to work with the Composite activity palette in an ActiveMatrix BusinessWorks™ process.

The process is designed with the following activities:

The following table lists and explains the activities in the process.

Activity	Description
Composite Tree	This activity demonstrates how to create two objects that have a hierarchical relationship. An Account object is created as a parent and a Contact object is created as a child of the Account object. Note that both the objects are created in one request.
Composite Batch	Two independent subrequests are configured in this activity. The two subrequests perform the following tasks: 1. Update the Account object that was created by the Composite Tree activity. 2. Use the Salesforce Search Query Builder to search for an object by its name and return the ID, industry, name, and phone.
ParseJSON	The output of the SOSL query in the Composite Batch activity is provided as input to the ParseJSON activity. The ParseJSON activity then generates the output in JSON format. This output can be used as an input to the next activity and the fields can be mapped.
Composite Dependent	The output of the ParseJSON activity is provided as an input to the Composite Dependent activity. Dependent subrequests are configured in this activity. The subrequests of this activity perform the following tasks: 1. Create an Opportunity for the Account object that was created by the Composite Tree activity. 2. Update the Opportunity. 3. Query the Opportunity by using Salesforce Query Builder. 4. Delete the Account object returned by the search query in Composite Batch activity. This is the Account object that was created by the Composite Tree activity.
ParseResponseJSON	This activity demonstrates that the output of a Composite activity can be used as an input.

CompositeActivityUsingJSONInputDemo Process

The CompositeActivityUsingJSONInputDemo process demonstrates how to work with the Composite activity palette using input in JSON format in an ActiveMatrix BusinessWorks™ process.

The process is designed with the following activities:
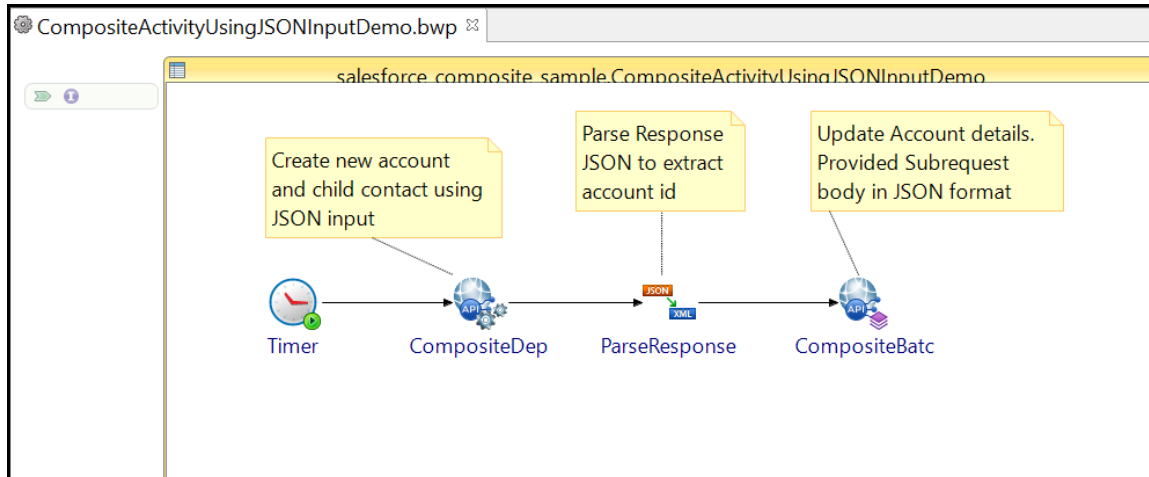
The following table explains the activities in the process:

Activity	Description
Composite Dependent	A parent object (Account) and its child object (Contact) are created by providing an existing JSON string in the **jsonBody** field.
ParseResponseJSON	The ParseResponseJSON activity demonstrates that the output of a Composite activity can be used as an input to another activity.
Composite Batch	The account ID from the parsed JSON file generated by the ParseResponseJSON activity is provided as an input to the Composite Batch activity. The subrequests of this activity are used to perform the following tasks: • Provide JSON input in a subrequest body • Delete the Account object that was created earlier by the Composite Dependent activity

Miscellaneous

The following topics have been discussed in this section:

- Properties Settings
- Time Zone Information
- Processing 16 Digits Numeric Data
- Salesforce Query Builder
- Salesforce Search Query Builder

Properties Settings

The properties settings explain the property settings you might have to configure for the product.

It includes the following sections:

- Retrying Function Setting
- Default Timeout Value Setting
- Proxy Settings

Retrying Function Setting

The retrying function is triggered when you retry connecting to the Salesforce.com server with the Salesforce configuration resource in ActiveMatrix BusinessWorks™ Plug-in for Salesforce.com. The retrying function is also triggered when you retry the actions in the Salesforce activities.

When you retry actions in the Salesforce activity for deployment, you have to set the following property lines in the *TIBCO_HOME*\bw\version_number\domains\defaultdomain\appnodes\defaultappspace \defaultappnode\config file:

```
com.tibco.plugin.salesforce.retry.rotationPolicy=2,5,20
com.tibco.plugin.salesforce.retry.rotationNumber=1
```

Property	Description
com.tibco.plugin.salesforce.retry.rotationPolicy	This property is used to set time interval between each retrying operation. It is applied on both Salesforce Activities and Salesforce Get Session activity.
	The default value in seconds is 2, 5, 20. If using this default value in our process, retrying operation is triggered three times. The activity waits for 2 seconds before retrying for the first time, 5 seconds before retrying for the second time, and 20 seconds before retrying for the third time..
com.tibco.plugin.salesforce.retry.rotationNumber	The property is used to set how many times the process defined in the com.tibco.plugin.salesforce.retry.rotationPolicy property is executed.

When you retry actions in the Salesforce activity in TIBCO Business Studio , you have to set the following property lines as shown in the following figure:

```
-Dcom.tibco.plugin.salesforce.retry.rotationPolicy=2,5,20
```

```
-Dcom.tibco.plugin.salesforce.retry.rotationNumber=1
```
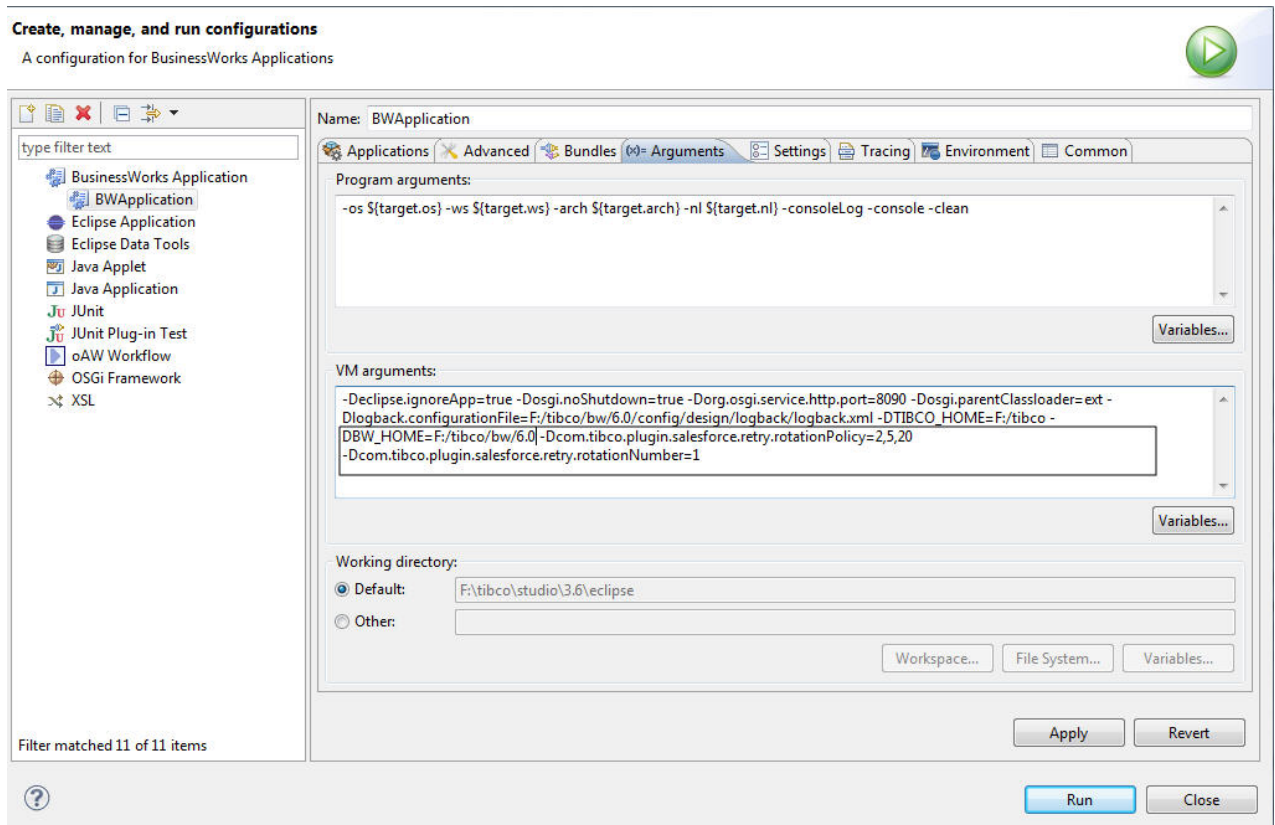
Default Timeout Value Setting

The timeout value specifies the number of milliseconds an internal API call waits before it is terminated.

When configuring the Salesforce activity used in a process, you have to set the timeout value for the activity. For example, for a Salesforce Query All activity, you have to set the timeout value as shown in the following figure:
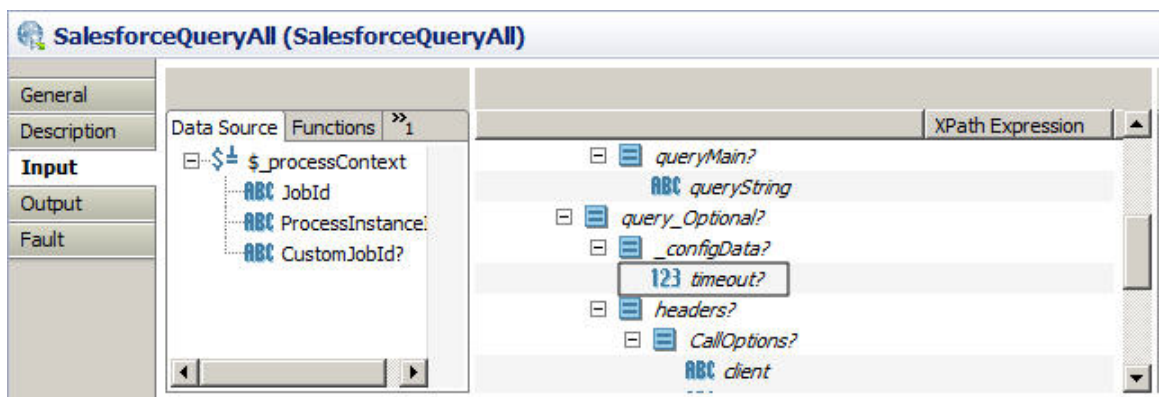
To conveniently configure the timeout value for the Salesforce activities running in TIBCO Business Studio, you can write the property line `-Dcom.tibco.plugin.salesforce.invoke.timeout=15000` as shown in the following figure:
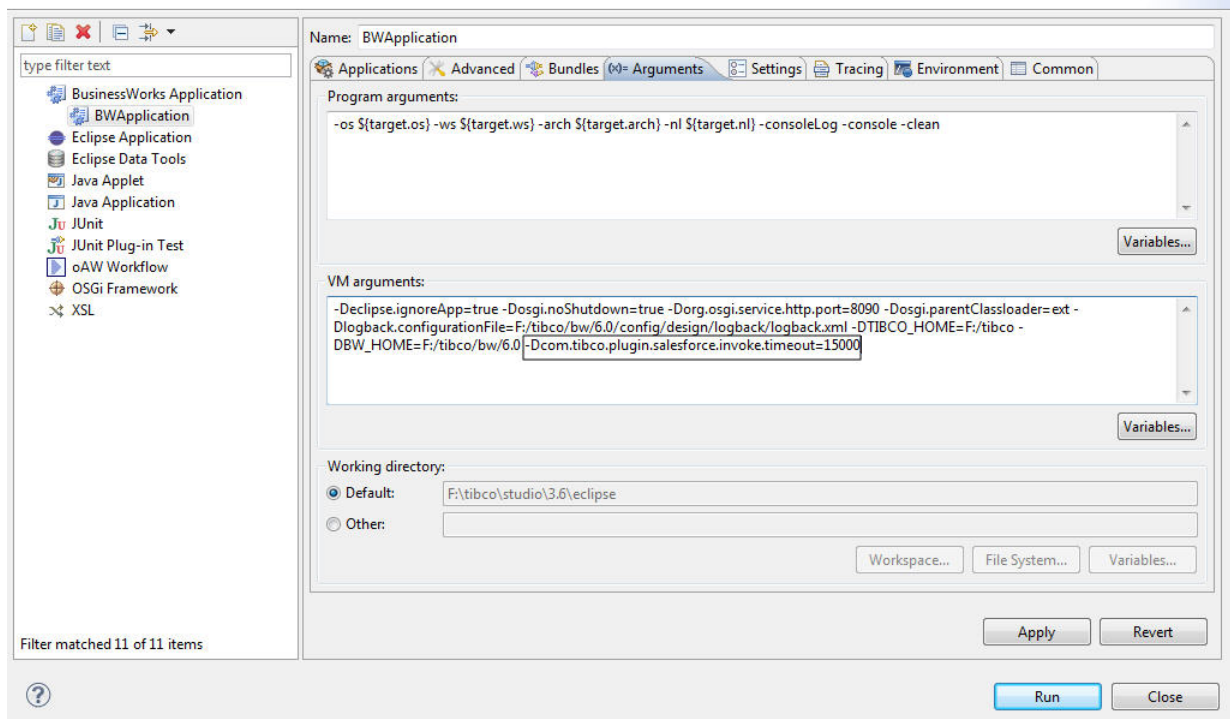
When you configure the timeout value in the Salesforce activity for deployment, you have to set the property `com.tibco.plugin.salesforce.invoke.timeout=15000` in the *TIBCO_HOME*\bw\domains \defaultdomain\appnodes\defaultappspace\defaultappnode\config file.

Property	Description
`com.tibco.plugin.salesforce.` `invoke.timeout`	This property is used to set the default timeout value for Salesforce activities. The default value in milliseconds is `15000`.

You can change the timeout value in this property line. This value is the default timeout value for each Salesforce activity.

Proxy Settings

The Plug-in HTTP proxy property is provided in ActiveMatrix BusinessWorks™ Plug-in for Salesforce.com.

When using Proxy, you have to operate proxy settings.

To conveniently configure Proxy in TIBCO Business Studio, you can set the properties as shown in the following figure:

`-Dcom.tibco.plugin.salesforce.proxyHost=192.168.54.11`

`-Dcom.tibco.plugin.salesforce.proxyPort=89`
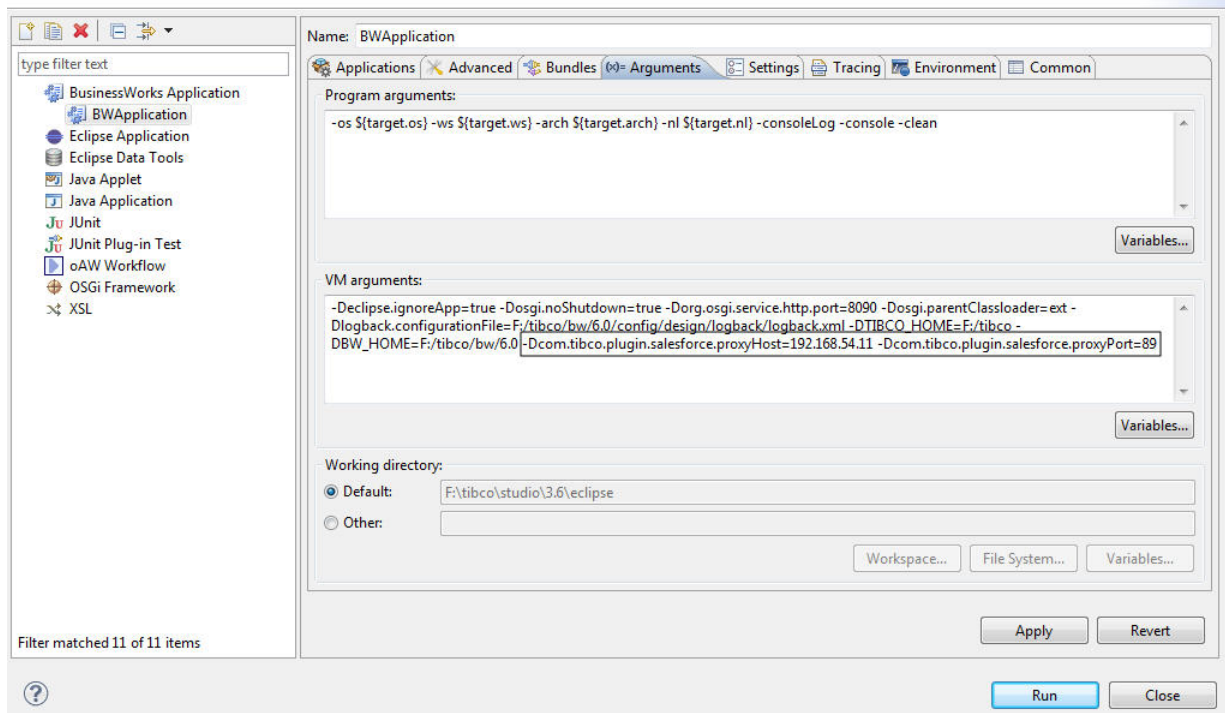
When you configure Proxy for deployment, you have to set these properties

```
com.tibco.plugin.salesforce.proxyHost=192.168.54.11
com.tibco.plugin.salesforce.proxyPort=89
```

in the *TIBCO_HOME*\bw*version_number*\domains\defaultdomain\appnodes\defaultappspace
\defaultappnode\config file.

Property	Description
`com.tibco.plugin.salesforce.proxyHost`	Required. The proxy host name or its IP address.
`com.tibco.plugin.salesforce.proxyPort`	Required. The port on which the proxy server is listening.
`com.tibco.plugin.salesforce.proxyUser`	Optional. The user name for logging on to the proxy server.
`com.tibco.plugin.salesforce.proxyPwd`	Optional. The password for logging on to the proxy server.
`com.tibco.plugin.salesforce.proxyTimeout`	Optional. The timeout value in minutes for an HTTP call. This property is to help avoid long wait times when attempting to retry through a proxy.

Alternately, you can also use the following proxy properties:

Property	Description
`http.proxyHost`	Required. The proxy host name or its IP address.
`http.proxyPort`	Required. The port on which the proxy server is listening.
`http.proxyUser`	Optional. The user name for logging on to the proxy server.
`http.proxyPassword`	Optional. The password for logging on to the proxy server.

When you use Proxy with the Salesforce Bulk palette and the proxy authentication using username and password, set the following properties in addition to the ones listed earlier:

- In the JVM properties

```
-Djdk.http.auth.tunneling.disabledSchemes=""
-Djdk.http.auth.proxying.disabledSchemes=""
```

- In the configuration file

```
jdk.http.auth.tunneling.disabledSchemes=""
jdk.http.auth.proxying.disabledSchemes=""
```

The same properties can be added in `TibcoBusinessStudio.ini` to configure proxy for GetMetadata, Refresh Metadata, Salesforce Test connection, and Composite subrequest handler.

Time Zone Information

TIBCO ActiveMatrix BusinessWorks Plug-in for Salesforce.com contains the following data types for defining the date or time data: date, dateTime, and time. Time zone information can be added to the dateTime and time data types.

Data Type	Data Format	Example	Description
date	yyyy-MM-dd	2009-01-31	Greenwich Mean Time (GMT) date.
			If you add the time zone information in the date data, TIBCO ActiveMatrix BusinessWorks Plug-in for Salesforce.com accepts it, but the time zone portion cannot be processed. For example, if the date data is 2009-01-31+08:00, it is truncated to 2009-01-31.
dateTime	yyyy-MM-ddTHH:mm:ss	2009-01-31T16:16:16	Greenwich Mean Time (GMT) datetime.
	yyyy-MM-ddTHH:mm:ssZ	2009-01-31T16:16:16Z	

Data Type	Data Format	Example	Description
	yyyy-MM-ddTHH:mm:ss+hh:mm	2009-01-31T16:16:16+08:00	The datetime data with the specified time zone.
	yyyy-MM-ddTHH:mm:ss-hh:mm	2009-01-31T16:16:16-08:00	
time	HH:mm:ss	16:16:16	Greenwich Mean Time (GMT) time.
	HH:mm:ssZ	16:16:16Z	
	HH:mm:ss+hh:mm	16:16:16+08:00	The time data with the specified time zone.
	HH:mm:ss-hh:mm	16:16:16-08:00	

Processing 16-Digit Numeric Data

You can input numeric data of a length greater than 16 digits in the Salesforce.com database.

For an application that is written in the JAVA language, data with a length greater than 16 digits is defined as the double type and its precision is changed in the application.

TIBCO ActiveMatrix BusinessWorks Plug-in for Salesforce.com is implemented using the JAVA language. When data with a length greater than 16 digits is processed between the plug-in and the Salesforce.com database, you might encounter the following situations:

- If the data is retrieved from the Salesforce.com database, their precision is rounded to 16 digits in the plug-in.

  Affected Salesforce activities are Salesforce Query All, Salesforce Retrieve All, and Salesforce Outbound Message Listener.

- If the data is sent to the Salesforce.com database, it is rounded to the 16-digit precision and then sent out.

  Affected Salesforce activities are Salesforce Create All, Salesforce Update All, and Salesforce Upsert All.

Salesforce Query Builder

The Salesforce Query Builder lets you create a query against a specified object.

Building a Query

You need to build a query when working with the Salesforce Query All activity from the Salesforce palette or the Salesforce Bulk Query activity from the Salesforce Bulk API palette. Click the **Query Builder** to open the wizard to build your query or clear the existing query with **Clear Query**.
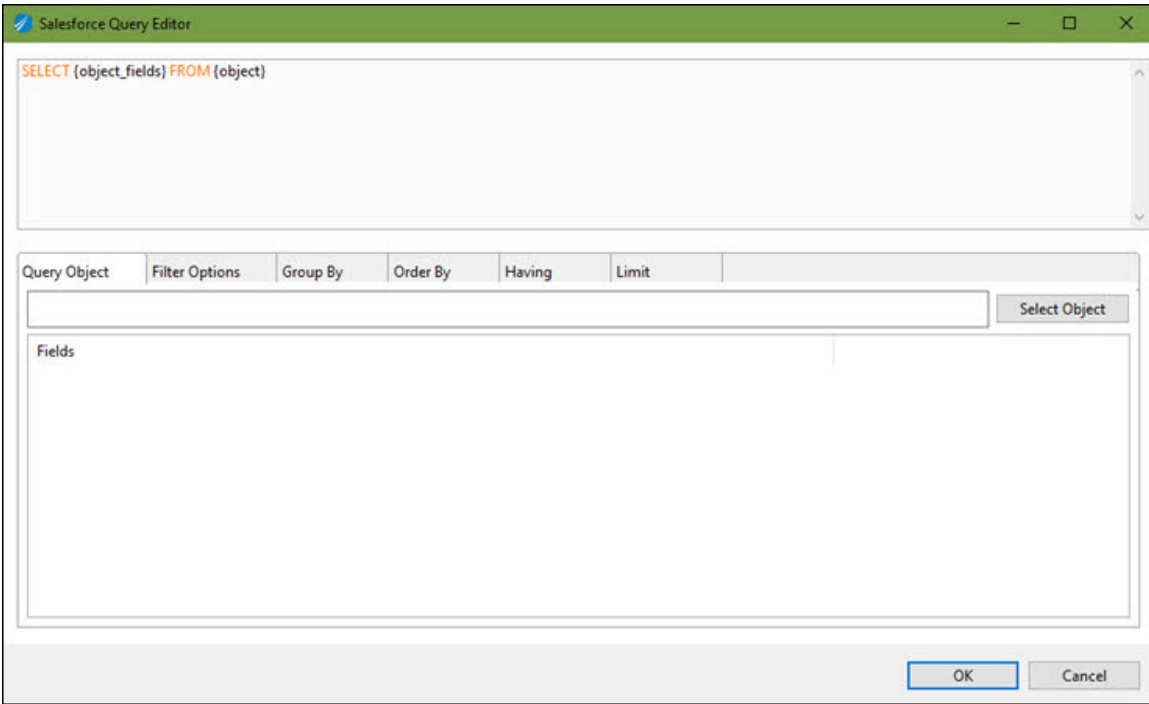
The following tabs are displayed on the Query Builder:

Tab	Description
Query Object	Select the Salesforce object and the fields on which you want to perform an operation. You can select only one object at a time.
Filter Options	Provide comparison operators in the query. You can combine multiple operators using the logical operators AND, OR. Select the field on which you want to apply operator for your query.
Group By	In the left hand panel you can select the fields on which you want to perform the group by operation. Using right hand panel you can select the aggregate and date functions.
Order By	Select the field by which you want to order the records.
Having	Specify the having clause for the fields selected on the **Group By** tab.
Limit	Provide limit and offset value for Salesforce records.

Salesforce Search Query Builder

The Salesforce Search Query Builder lets you construct SOSL queries to perform text-based search queries.

Building a Search Query

When creating a Salesforce Composite Batch subrequest, if you select the resource as search, you can use Salesforce Search Query Builder to build the search query. Click **Search Query Builder** to open the wizard to build your search query.
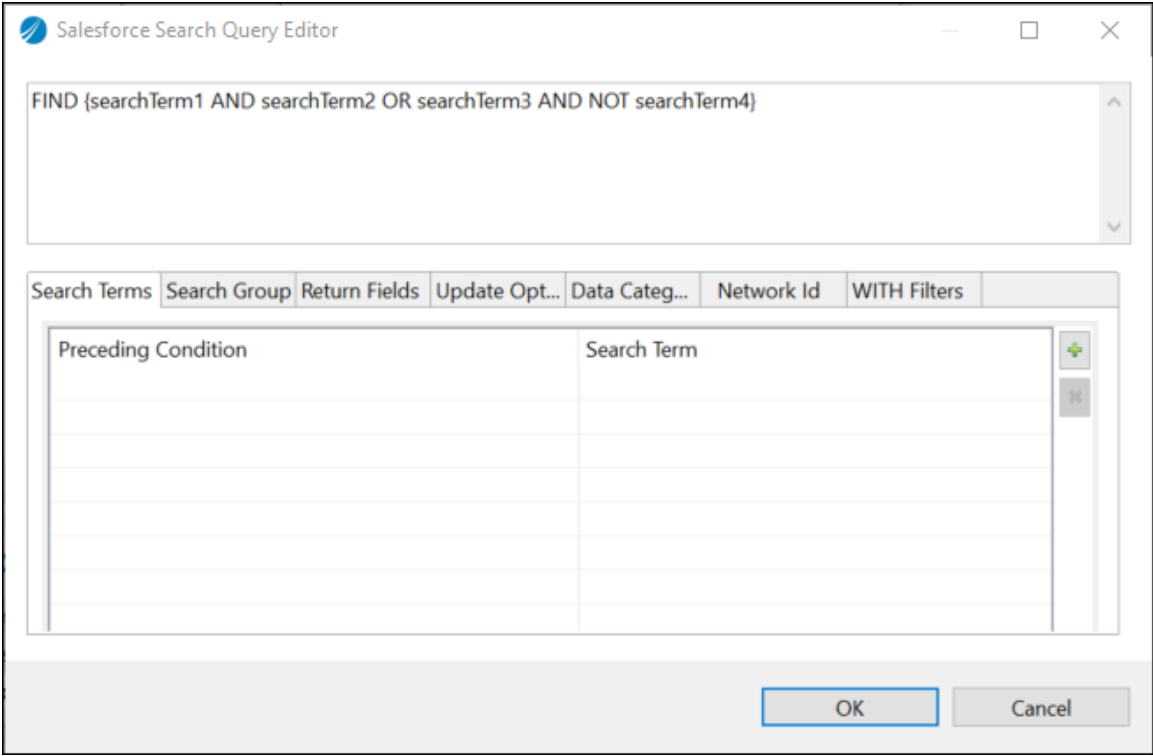
The following tabs are displayed on the Search Query Builder:

Tabs	Description
Search Terms	Specify the combination of words or phrases to search for.
Search Group	Select the type of text fields to search the words or phrases specified in the Search Terms.
Return Fields	Specify the object and the fields to be returned in the search result. You can use the Salesforce Query Builder to build the object query by clicking the Query Builder icon.
Update Options	Select either Tracking or Viewstat option to enable updating Tracking or Viewstat for Salesforce Knowledge Articles.

Tabs	Description
Data Category	Use this to add data category filter to your search query. The Data Category Group, Preceding Condition, and Data Category are populated in the drop down fields from the Salesforce server.
Network Id	You can search for community users and feeds by using the WITH NETWORK optional clause on a SOSL query.
WITH Filters	Use this to provide the following filters: • Division Filter • Highlight • Metadata • Snippet Target Length • Spell Correction • Limit

For more information about the fields, see the Salesforce documentation.

Troubleshooting

When you encounter problems with the project, ensure that you have cleaned up the project.

If errors occur when you run a process in TIBCO Business Studio, you can do a cleaning first. Cleaning deletes all the old files and reorganizes the project.

Procedure

1. Right-click the project in the Project Explorer view and click **Refresh**.
2. Click **Project** > **Clean** to start the cleaning.

Managing Logs

When an error occurs, you can check logs to trace and troubleshoot the plug-in exceptions. By default, the error logs are displayed in Console view after you run a process in debug mode.

Different log levels correspond to different messages, as described in Log Levels. You can change the log level of the plug-in to trace different messages. For more information about setting up the console log level, see Setting Up Log Levels.

You can export logs to a file by modifying the `logback.xml` file. You can also export logs of the defined log levels to a file. For more information about exporting logs to a file, see Exporting Logs.

Log Levels

Different log levels include different information.

The plug-in supports the following log levels:

Log Level	Description
Trace	Includes all information regarding the running process.
Debug	Indicates a developer-defined tracing message.
Info	Indicates normal plug-in operations. No action is required. A tracing message tagged with Info indicates that a significant processing step is reached, and logged for tracking or auditing purposes. Only info messages preceding a tracking identifier are considered as significant steps.
Warn	Indicates that an abnormal condition occurred. Processing continues, but for best practice, you can contact the administrator to investigate it.
Error	Indicates that an unrecoverable error occurred. Depending on the severity of the error, the plug-in might continue with the next operation or might stop.

Setting Up Log Levels

You can debug a process and view the logs in Console view.

For Salesforce palette:

```
<logger name="com.tibco.bw.palette.salesforce.runtime">
   <level value="DEBUG"/>
</logger>
```

For Salesforce Subscriber activity:

```
<logger name="com.tibco.bw.palette.salesforce.streaming.runtime">
   <level value="DEBUG"/>
</logger>
```

For Salesforce Bulk API palette:

```
<logger name="com.tibco.bw.palette.sfbulk.runtime">
   <level value="DEBUG"/>
</logger>
```

For Salesforce Bulk 2.0 API

```
<logger name="com.tibco.bw.palette.sfbulk2.runtime">
<level value="DEBUG"/>
</logger>
```

For Salesforce Composite API palette:

```
<logger name="com.tibco.bw.palette.salesforce.composite.runtime">
    <level value="DEBUG"/>
</logger>
```

The `level` tag defines the log level. The value can be `TRACE`, `DEBUG`, `INFO`, `WARN`, and `ERROR`.

After adding the previous lines to the `logback.xml` file, the log status of all activities is displayed in the Console view. If you want to display a specified activity log status, choose the following corresponding lines:

For Salesforce palette:

```
com.tibco.bw.palette.salesforce.runtime.SalesforceQueryActivity
com.tibco.bw.palette.salesforce.runtime.SalesforceCreateActivity
com.tibco.bw.palette.salesforce.runtime.SalesforceUpdateActivity
com.tibco.bw.palette.salesforce.runtime.SalesforceRetrieveActivity
com.tibco.bw.palette.salesforce.runtime.SalesforceUpsertActivity
com.tibco.bw.palette.salesforce.runtime.SalesforceDeleteActivity
com.tibco.bw.palette.salesforce.runtime.SalesforceGetSessionActivity
com.tibco.bw.palette.salesforce.streaming.runtime.SalesforceStreamingEventSource
com.tibco.bw.palette.salesforce.runtime.eventsource.SalesforceOutboundListenerEventSour
ce
```

For Salesforce Bulk API palette:

```
com.tibco.bw.palette.sfbulk.runtime.SalesforceBulkOperationActivity
com.tibco.bw.palette.sfbulk.runtime.SalesforceBulkQueryActivity
com.tibco.bw.palette.sfbulk.runtime.SalesforceCheckStatusActivity
com.tibco.bw.palette.sfbulk.runtime.SalesforceGetResultActivity
```

For Composite API Palette

```
com.tibco.bw.palette.salesforce.composite.runtime.CompositeTreeAsynchronousActivity
com.tibco.bw.palette.salesforce.composite.runtime.CompositeDependentAsynchronousActivit
y
com.tibco.bw.palette.salesforce.composite.runtime.CompositeBatchAsynchronousActivity
```

Add for Salesforce Bulk API 2.0 :

```
com.tibco.bw.palette.sfbulk2.runtime.SfbulkOperationActivity
com.tibco.bw.palette.sfbulk2.runtime.SfbulkQueryActivity
com.tibco.bw.palette.sfbulk2.runtime.SfbulkGetResultActivity
com.tibco.bw.palette.sfbulk2.runtime.SfbulkCheckStatusActivity
com.tibco.bw.palette.sfbulk2.runtime.SfbulkCloseJobActivity
```

For example, use the following code lines to display the log status of the Salesforce Query All activity:

```
<logger name="com.tibco.bw.palette.salesforce.runtime.SalesforceQueryActivity">
    <level value="DEBUG"/>
</logger>
```

> When deploying the application in TIBCO Enterprise Administrator, you can navigate to the *TIBCO_HOME*
> \bw\version_number\domains\defaultdomain\appnodes\defaultappspace\defaultappnode directory
> to find the `logback.xml` file.

Exporting Logs

You can print the log to a file by adding FILE appender in the `logback.xml` file.

Procedure

1. Navigate to the *TIBCO_HOME*\bw*version_number*\config\design\logback directory and open the
    `logback.xml` file.

    > When deploying the application in TIBCO Enterprise Administrator, you can navigate to the
    > *TIBCO_HOME*\bw*version_number*\domains\defaultdomain\appnodes\defaultappspace
    > \defaultappnode directory to find the `logback.xml` file.

2. Add the following node to specify the file location.

```
<appender name="FILE" class="ch.qos.logback.core.FileAppender">
    <file>c:/bw6-salesforce.log</file>
        <encoder>
          <pattern>%d{HH:mm:ss.SSS} [%thread] %-5level %logger{36}-%msg%n</pattern>
        </encoder>
</appender>
```

The `file` tag defines the location to which the log is exported and the value is the absolute path of the file that is detailed to the file name.

3. Add the following node to specify the log level.

```
<logger name="com.tibco.bw.palette.salesforce.runtime">
    <appender-ref ref="FILE"/>
    <level value="ERROR"/>
</logger>
```

The `level` tag defines the log level and the value can be TRACE, DEBUG, INFO, WARN and ERROR.

4. Save the file.

Error Codes

The error codes on the **Fault** tab for each activity are listed with the corresponding description and resolution.

Error codes for the following palettes are listed.

- Salesforce Palette
- Salesforce Bulk API Palette
- Salesforce Bulk API 2.0 Palette
- Salesforce Composite Palette

Salesforce Palette

Error Code	Role	Category	Description	Resolution
TIBCO-BW-PALETTE-SALESFORCE-100001	traceRole	BW-Plugin	Request SOAP message sent for operation {0}: \n {1}	None
TIBCO-BW-PALETTE-SALESFORCE-100002	traceRole	BW-Plugin	Response SOAP message received for operation {0}: \n {1}	None
TIBCO-BW-PALETTE-SALESFORCE-200001	debugRole	BW-Plugin	{0}	None
TIBCO-BW-PALETTE-SALESFORCE-200002	debugRole	BW-Plugin	The operation name has been changed from [{0}] to [{1}].	None
TIBCO-BW-PALETTE-SALESFORCE-200003	debugRole	BW-Plugin	The following Request SOAP Message is sent \n {0}	None
TIBCO-BW-PALETTE-SALESFORCE-300001	infoRole	BW-Plugin	The session for User {0} timed out. Old sessionId is {1}.	None
TIBCO-BW-PALETTE-SALESFORCE-300002	infoRole	BW-Plugin	The session for User {0} was refreshed. Old sessionId is {1}.	None

Error Code	Role	Category	Description	Resolution
TIBCO-BW-PALETTE-SALESFORCE-3000 03	infoRole	BW-Plugin	User {0} is logging in.	None
TIBCO-BW-PALETTE-SALESFORCE-3000 04	infoRole	BW-Plugin	This is the {1} time(s) during the rotation {2} for user {0} trying to login.	None
TIBCO-BW-PALETTE-SALESFORCE-3000 05	infoRole	BW-Plugin	Login failed for User {0}.	None
TIBCO-BW-PALETTE-SALESFORCE-3000 06	infoRole	BW-Plugin	Login succeeded for User {0}. SessionId is [{1}]	None
TIBCO-BW-PALETTE-SALESFORCE-3000 07	infoRole	BW-Plugin	Get session for User {0} immediately. SessionId is [{1}]	None
TIBCO-BW-PALETTE-SALESFORCE-3000 08	infoRole	BW-Plugin	Mechanism Retry of Getting Session makes User {0} sleep for {1} seconds.	None
TIBCO-BW-PALETTE-SALESFORCE-3000 09	infoRole	BW-Plugin	Request SOAP Message Sent \n {0}	None
TIBCO-BW-PALETTE-SALESFORCE-3000 10	infoRole	BW-Plugin	Response SOAP Message Received \n {0}	None
TIBCO-BW-PALETTE-SALESFORCE-3000 11	infoRole	BW-Plugin	Start to operate [{0}]. Process ID: [{1}]. Engine Name: [{2}].	None
TIBCO-BW-PALETTE-SALESFORCE-3000 12	infoRole	BW-Plugin	Renewed session in {0} activity.	None

Error Code	Role	Category	Description	Resolution
TIBCO-BW-PALETTE-SALESFORCE-300013	infoRole	BW-Plugin	Ready to execute the operation [{0}].	None
TIBCO-BW-PALETTE-SALESFORCE-300014	infoRole	BW-Plugin	Ready to execute the operation [{0}]. The related input count is {1}.	None
TIBCO-BW-PALETTE-SALESFORCE-300015	infoRole	BW-Plugin	Trying to refresh the session.	None
TIBCO-BW-PALETTE-SALESFORCE-300016	infoRole	BW-Plugin	Refreshed the session. Be ready to retry the operation.	None
TIBCO-BW-PALETTE-SALESFORCE-300017	infoRole	BW-Plugin	Ready to execute the operation [Query], the query string is [{0}].	None
TIBCO-BW-PALETTE-SALESFORCE-300018	infoRole	BW-Plugin	Be ready to execute the operation [QueryMore], the query locator is [{0}].	None
TIBCO-BW-PALETTE-SALESFORCE-300019	infoRole	BW-Plugin	Processing in subsets. Batch size is [{0}], subset size is [{1}].	None
TIBCO-BW-PALETTE-SALESFORCE-300020	infoRole	BW-Plugin	The configured BatchSize is [{0}].	None
TIBCO-BW-PALETTE-SALESFORCE-300021	infoRole	BW-Plugin	Be ready to process the operation in subsets for [{0}] activity. The configured BatchSize is [{1}], and the SubsetSize is [{2}].	None
TIBCO-BW-PALETTE-SALESFORCE-300022	infoRole	BW-Plugin	The end of the [{0}] activity.	None

Error Code	Role	Category	Description	Resolution
TIBCO-BW-PALETTE-SALESFORCE-300023	infoRole	BW-Plugin	Retry to execute the [{0}] operation in [{1}] seconds.	None
TIBCO-BW-PALETTE-SALESFORCE-300024	infoRole	BW-Plugin	This is the [{0}] time(s) during the rotation [{1}] for trying [{2}] operation.	None
TIBCO-BW-PALETTE-SALESFORCE-300025	infoRole	BW-Plugin	Processing in subsets is finished for [{0}] activity.	None
TIBCO-BW-PALETTE-SALESFORCE-300026	infoRole	BW-Plugin	Be ready to process the operation for [{0}] activity.	None
TIBCO-BW-PALETTE-SALESFORCE-300027	infoRole	BW-Plugin	Processing the operation is finished for [{0}] activity.	None
TIBCO-BW-PALETTE-SALESFORCE-300028	infoRole	BW-Plugin	Recoverable errors returned with the records. Find the information of the related records in the log.	None
TIBCO-BW-PALETTE-SALESFORCE-300029	infoRole	BW-Plugin	Line separator for info logs. ---	None
TIBCO-BW-PALETTE-SALESFORCE-300030	infoRole	BW-Plugin	The results have been returned successfully. No error occurs in the returned data.	None
TIBCO-BW-PALETTE-SALESFORCE-300032	infoRole	BW-Plugin	End of executing a batch of the operation for [{0}] activity. [{1}] sObject(s) have been returned.	None
TIBCO-BW-PALETTE-SALESFORCE-300033	infoRole	BW-Plugin	Salesforce Outbound Message Listener on [{0}] was initialized successfully.	None

Error Code	Role	Category	Description	Resolution
TIBCO-BW-PALETTE-SALESFORCE-3000 36	infoRole	BW-Plugin	Be ready to execute the operation [QueryMore] if any nested sObject exists.	None
TIBCO-BW-PALETTE-SALESFORCE-3000 37	infoRole	BW-Plugin	End of executing a batch of the operation for the nested node [{2}] of [{0}] activity. [{1}] sObject(s) have been returned.	None
TIBCO-BW-PALETTE-SALESFORCE-3000 38	infoRole	BW-Plugin	The invocation timeout value is {0} ms.	None
TIBCO-BW-PALETTE-SALESFORCE-3000 39	infoRole	BW-Plugin	The timeout value for this call are [{0}] milliseconds.	None
TIBCO-BW-PALETTE-SALESFORCE-3000 40	infoRole	BW-Plugin	The Salesforce Outbound Message Listener activity received a message whose ServiceEntry URL is [{0}]. Sending an Ack: [{1}].	None
TIBCO-BW-PALETTE-SALESFORCE-3000 41	infoRole	BW-Plugin	The Salesforce Outbound Message Listener activity received a message whose ServiceEntry URL is [{0}].	None
TIBCO-BW-PALETTE-SALESFORCE-3000 42	infoRole	BW-Plugin	Proxy setting in use, the proxyHost:[{0}], proxyPort:[{1}], proxyUser:[{2}], proxyPwd:[{3}], proxyTimeout(min): [{4}].	None
TIBCO-BW-PALETTE-SALESFORCE-3000 43	infoRole	BW-Plugin	End of executing a batch of operation for activity [{0}], but no sObject returned.	None

Error Code	Role	Category	Description	Resolution
TIBCO-BW-PALETTE-SALESFORCE-300044	infoRole	BW-Plugin	End of executing a batch of operation for the node [{1}] of [{0}] activity, but no sObject is returned.	None
TIBCO-BW-PALETTE-SALESFORCE-400001	warnRole	BW-Plugin	The batchSize was not assigned with a valid value. It is given a default value.	None
TIBCO-BW-PALETTE-SALESFORCE-400002	warnRole	BW-Plugin	The input of 'salesforceConnection' is empty. The operation cannot be retried when an invalid session exception occurred.	None
TIBCO-BW-PALETTE-SALESFORCE-400003	warnRole	BW-Plugin	The configured BatchSize is [{0}] that is fixed to the minimum (200) or maximum (2000) BatchSize.	None
TIBCO-BW-PALETTE-SALESFORCE-400004	warnRole	BW-Plugin	The configured SubsetSize is [{0}] that is fixed to the minimum (1) or maximum (10) SubsetSize.	None
TIBCO-BW-PALETTE-SALESFORCE-400005	warnRole	BW-Plugin	The configured BatchSize for [{0}] operation is [{1}] that is fixed to the minimum (200) or maximum (2000) BatchSize.	None
TIBCO-BW-PALETTE-SALESFORCE-400006	warnRole	BW-Plugin	The configured SubsetSize for [{0}] operation is [{1}] that is fixed to the minimum (1) or maximum (10) SubsetSize.	None
TIBCO-BW-PALETTE-SALESFORCE-400007	warnRole	BW-Plugin	The [{0}] activity ends up with the error(s).	None

Error Code	Role	Category	Description	Resolution
TIBCO-BW-PALETTE-SALESFORCE-400008	warnRole	BW-Plugin	No error information is available in the returned data.	None
TIBCO-BW-PALETTE-SALESFORCE-400009	warnRole	BW-Plugin	End of executing a batch of operation for activity [{0}], but no sObject returned.	None
TIBCO-BW-PALETTE-SALESFORCE-400010	warnRole	BW-Plugin	No input data under the node [{0}] at run time for activity [{1}] is available. An empty result is returned directly.	None
TIBCO-BW-PALETTE-SALESFORCE-400011	warnRole	BW-Plugin	The session refreshing operation could not be performed because the external session configuration has been used.	None
TIBCO-BW-PALETTE-SALESFORCE-400012	warnRole	BW-Plugin	End of executing a batch of operation for the node [{1}] of [{0}] activity, but no sObject is returned.	None
TIBCO-BW-PALETTE-SALESFORCE-500001	errorRole	BW-Plugin	Cannot find shared Salesforce configuration: {0}	Check the configuration of the Salesforce connection and reconfigure it.
TIBCO-BW-PALETTE-SALESFORCE-500002	errorRole	BW-Plugin	Error occurred while logging in: {0}	Check whether the network is connected and the user name and password are correct.
TIBCO-BW-PALETTE-SALESFORCE-500003	errorRole	BW-Plugin	FatalException: Failed to execute the operation after retrying [{0}] time(s).	Check the network configuration.
TIBCO-BW-PALETTE-SALESFORCE-500004	errorRole	BW-Plugin	OperationException: Failed to execute the operation. Check the input configuration. {0} Detailed information: [{1}] Other information: [{2}]	Check the input configuration.

Error Code	Role	Category	Description	Resolution
TIBCO-BW-PALETTE-SALESFORCE-500005	errorRole	BW-Plugin	FatalException: Failed to renew the session in {0} activity.	Check the input of the salesforceConnection field.
TIBCO-BW-PALETTE-SALESFORCE-500006	errorRole	BW-Plugin	Exception occurred while executing the operation. Exception type: [{0}] Exception message: [{1}] Exception detail: [{2}] Other information: [{3}]	None
TIBCO-BW-PALETTE-SALESFORCE-500007	errorRole	BW-Plugin	Network problem occurred. The server is currently unavailable or the session has timed out.	Check the network configuration.
TIBCO-BW-PALETTE-SALESFORCE-500008	errorRole	BW-Plugin	The [{0}] operation is temporarily unavailable.	Contact Salesforce.com support.
TIBCO-BW-PALETTE-SALESFORCE-500009	errorRole	BW-Plugin	An unrecoverable error occurred while executing the {0} operation.	Check the input configuration of the activity.
TIBCO-BW-PALETTE-SALESFORCE-500010	errorRole	BW-Plugin	An unrecoverable error occurred while inputting the invalid data for the {0} activity. Check the input configuration of the activity.	Check the input configuration of the activity.
TIBCO-BW-PALETTE-SALESFORCE-500011	errorRole	BW-Plugin	ConnectionException: Failed to execute the operation.	Check the network configuration.
TIBCO-BW-PALETTE-SALESFORCE-500012	errorRole	BW-Plugin	Detailed information about this unknown exception: StackTrace: {0} Exception Message: [{1}]. FullClass [{2}].	Check the input configuration of the activity or contact TIBCO support.
TIBCO-BW-PALETTE-SALESFORCE-500013	errorRole	BW-Plugin	Data was not returned in limited operation time while executing [{0}] operation.	Change the timeout value or check the connection of the network.

Error Code	Role	Category	Description	Resolution
TIBCO-BW-PALETTE-SALESFORCE-500014	errorRole	BW-Plugin	Application exception occurred while handling an error.	Check the input configuration of the activity or contact TIBCO support.
TIBCO-BW-PALETTE-SALESFORCE-500015	errorRole	BW-Plugin	FatalException: Invalid session error occurred. Since an external session ID was used, the refreshing session mechanism cannot be performed. The external session ID is {0}.	Make sure the external session ID is valid.
TIBCO-BW-PALETTE-SALESFORCE-500016	errorRole	BW-Plugin	ApplicationException: The operation failed. {0} Detailed information: [{1}]. Other information: [{2}].	Check the input configuration of the activity or contact TIBCO support.
TIBCO-BW-PALETTE-SALESFORCE-500017	errorRole	BW-Plugin	UnknownException: The operation failed. {0}	Check the input configuration of the activity or contact TIBCO support.
TIBCO-BW-PALETTE-SALESFORCE-500018	errorRole	BW-Plugin	ConnectionException: The operation was failed {0} Detailed information: [{1}] Other information: [{2}]	Check the network connection.
TIBCO-BW-PALETTE-SALESFORCE-500019	errorRole	BW-Plugin	TimeoutException: The timeout value is [{0}]. Exception Message: [{1}]	Change the timeout value or check the connection of the network.
TIBCO-BW-PALETTE-SALESFORCE-500020	errorRole	BW-Plugin	The provided certification is invalid or the selected encryption strength might not match your policy file.	Make sure the Salesforce Resources folder has been imported and Java Cryptography Extension (JCE) has been installed.
TIBCO-BW-PALETTE-SALESFORCE-500021	errorRole	BW-Plugin	ApplicationException: The required field [{0}] is empty. A valid value is required.	Make sure that a valid value is assigned in the required field.

Error Code	Role	Category	Description	Resolution
TIBCO-BW-PALETTE-SALESFORCE-500022	errorRole	BW-Plugin	OperationException: The returned sObject contained an error information. statusCode: [{0}] message: [{1}] sObject ID in the returned data: [{2}] fields: [{3}] the sObject ID in the input: [{4}]	Check the input configuration of sObjects.
TIBCO-BW-PALETTE-SALESFORCE-500023	errorRole	BW-Plugin	The operation cannot be continued because some unrecoverable errors occurred. statusCode: [{0}] message: [{1}] Check the error message that was generated from the activity output.	Check the error message that was generated from the activity output.
TIBCO-BW-PALETTE-SALESFORCE-500024	errorRole	BW-Plugin	OperationException: The returned sObject contains an error information. statusCode: [{0}] message: [{1}] sObject ID in the returned data: [{2}] fields: [{3}].	Check the input configuration of sObjects.
TIBCO-BW-PALETTE-SALESFORCE-500025	errorRole	BW-Plugin	OperationException: The returned sObject has error. statusCode: [{0}]. message: [{1}]. sObject ID in the returned data: [{2}]. fields: [{3}]. value assigned for externalIDFieldName: [{4}]	Check the input configuration of sObjects.
TIBCO-BW-PALETTE-SALESFORCE-500026	errorRole	BW-Plugin	Error occurred while executing the Salesforce SOAP method: The method is {0}, details - {1}.	Check the input configuration of the activity.
TIBCO-BW-PALETTE-SALESFORCE-500028	errorRole	BW-Plugin	Failed to retry the operation [{0}].	Check the input configuration of the activity.

Error Code	Role	Category	Description	Resolution
TIBCO-BW-PALETTE-SALESFORCE-500029	errorRole	BW-Plugin	Failed to retry the operation in a rotation.	Check the input configuration of the activity.
TIBCO-BW-PALETTE-SALESFORCE-500030	errorRole	BW-Plugin	Unrecoverable errors returned with the records. You can find the information of the related records in the log.	Check the input configuration of the activity.
TIBCO-BW-PALETTE-SALESFORCE-500031	errorRole	BW-Plugin	The required field 'externalIdFieldName' was empty. Could not do retrying for the Create All activity.	Fill in the value in the externalIdFieldName field.
TIBCO-BW-PALETTE-SALESFORCE-500032	errorRole	BW-Plugin	Failed to retry the operation for all [{0}] rotation(s).	Check the input configuration of the activity.
TIBCO-BW-PALETTE-SALESFORCE-500033	errorRole	BW-Plugin	Error was kept in the returned sObject. The index number in the batch: [{0}], statusCode: [{1}], message: [{2}], sObject ID in the returned data (if available): [{3}], fields (if available): [{4}].	Check the error message that was generated from the activity output.
TIBCO-BW-PALETTE-SALESFORCE-500034	errorRole	BW-Plugin	Error occurred on the Outbound Message Listener activity: {0}.	Check the input configuration of the activity.
TIBCO-BW-PALETTE-SALESFORCE-500035	errorRole	BW-Plugin	Unrecoverable error was kept in the returned sObject. The index number in the batch: [{0}], statusCode: [{1}], message: [{2}], sObject ID in the returned data (if available): [{3}], fields (if available): [{4}].	Check the input configuration of the activity.

Error Code	Role	Category	Description	Resolution
TIBCO-BW-PALETTE-SALESFORCE-5000 36	errorRole	BW-Plugin	Recoverable error was kept in the returned sObject. The index number in the batch: [{0}], statusCode: [{1}], message: [{2}], sObject ID in the returned data (if available): [{3}], fields (if available): [{4}].	Check the input configuration of the activity.
TIBCO-BW-PALETTE-SALESFORCE-5000 37	errorRole	BW-Plugin	Common error was kept in the returned sObject. The index number in the batch: [{0}], statusCode: [{1}], message: [{2}], sObject ID in the returned data (if available): [{3}], fields (if available): [{4}].	Check the input configuration of the activity.
TIBCO-BW-PALETTE-SALESFORCE-5000 38	errorRole	BW-Plugin	The operation failed. Error message: {0}.	Check the input configuration of the activity.
TIBCO-BW-PALETTE-SALESFORCE-5000 39	errorRole	BW-Plugin	Could not find an available session for the operation [{0}]. Check the configuration of the Salesforce connection.	Check the configuration of the Salesforce connection.
TIBCO-BW-PALETTE-SALESFORCE-5000 40	errorRole	BW-Plugin	The WSDL was NOT specified.	Configure an Outbound Message WSDL generated from the Salesforce.com server.
TIBCO-BW-PALETTE-SALESFORCE-5000 41	errorRole	BW-Plugin	Keystore File was NOT specified.	Configure Keystore File.
TIBCO-BW-PALETTE-SALESFORCE-5000 42	errorRole	BW-Plugin	Trusted Certificates Folder was NOT specified.	Configure Trusted Certificates Folder.
TIBCO-BW-PALETTE-SALESFORCE-5000 43	errorRole	BW-Plugin	Failed to start outbound listener because of[{0}]	None

Error Code	Role	Category	Description	Resolution
TIBCO-BW-PALETTE-SALESFORCE-500200	errorRole	BW-Plugin	Generic Plug-in error	Attempt resolution based on the error message. If that fails enable debug logging. If that fails contact TIBCO support.
TIBCO-BW-PALETTE-SALESFORCE-500201	errorRole	BW-Plugin	Subscriber failed to login for reason	Ensure that the connection is correctly configured. Enable debug logging for more information.
TIBCO-BW-PALETTE-SALESFORCE-500202	errorRole	BW-Plugin	Subscriber failed to start for reason	Ensure that the connection is correctly configured. Enable debug logging for more information.
TIBCO-BW-PALETTE-SALESFORCE-500203	errorRole	BW-Plugin	Subscriber failed to stop for reason	Resolve based on this message and if that fails enable debug logging for more information. If that fails contact TIBCO support.
TIBCO-BW-PALETTE-SALESFORCE-500204	errorRole	BW-Plugin	Subscriber failed to parse the message for reason	Resolve based on this message and if that fails enable debug logging for more information. If that fails contact TIBCO support.
TIBCO-BW-PALETTE-SALESFORCE-500205	errorRole	BW-Plugin	Handshake failed for the URL	Ensure connection and WSDL is correctly configured. Enable debug logging for more information. If that fails contact TIBCO support.
TIBCO-BW-PALETTE-SALESFORCE-500206	errorRole	BW-Plugin	Buffering capacity exceeded.	Ensure that the following Salesforce streaming API property is set to an appropriate value: `com.tibco.plugin.salesforce.streaming.buffer`

Salesforce Bulk API Palette

Error Code	Role	Category	Description	Resolution
TIBCO-BW-PALETTE-SFBULK-100004	errorRole	BW-Plugin	Salesforce Connection is not provided in the activity	Provide Salesforce Connection Shared Resource on the **General** tab
TIBCO-BW-PALETTE-SFBULK-100005	errorRole	BW-Plugin	The sObjects element in the input was not correctly mapped	Map the sObject element in the input by a valid Salesforce Object substitution
TIBCO-BW-PALETTE-SFBULK-100006	errorRole	BW-Plugin	Error in parsing the files provided	The file provided as input is not present, is incorrectly formatted, or not readable
TIBCO-BW-PALETTE-SFBULK-100007	errorRole	BW-Plugin	Could not access the WSDL file provided	The WSDL file provided could not be accessed
TIBCO-BW-PALETTE-SFBULK-100008	errorRole	BW-Plugin	File access, parse error	One of the files provided could not be accessed or is not present
TIBCO-BW-PALETTE-SFBULK-100009	errorRole	BW-Plugin	Error logging into Salesforce instance	Check the credentials provided and the user account
TIBCO-BW-PALETTE-SFBULK-100010	errorRole	BW-Plugin	External session details are not provided	External session is set to true but sessionId and serverUrl are not provided or are empty
TIBCO-BW-PALETTE-SFBULK-100011	errorRole	BW-Plugin	Error occurred during firing jersey request to Salesforce Rest API	Check network connection and manually ping Salesforce Rest API
TIBCO-BW-PALETTE-SFBULK-100012	errorRole	BW-Plugin	Content type not provided for operation	Provide content type either on the **General** tab, in the input section, or in the substitution of sObject element

Error Code	Role	Category	Description	Resolution
TIBCO-BW-PALETTE-SFBULK-100013	errorRole	BW-Plugin	Object type mismatch between object type provided and substituted	The object type provided on the **General** tab or the **Input** tab does not match with the object type substituted for the sObject element. Make object type consistent.
TIBCO-BW-PALETTE-SFBULK-100014	errorRole	BW-Plugin	Object type not provided for Bulk operation	Provide object type either on the **General** tab, in the input section, or in the substitution of sObject element
TIBCO-BW-PALETTE-SFBULK-100015	errorRole	BW-Plugin	Error creating create job request for bulk operation	Fields such as operation and externalIdFieldName are not provided. Provide all necessary and required fields
TIBCO-BW-PALETTE-SFBULK-100016	errorRole	BW-Plugin	Error creating add batch request for bulk operation	Required fields are missing or not able to submit add batch request to Salesforce server. Check network, required fields, files provided.
TIBCO-BW-PALETTE-SFBULK-100017	errorRole	BW-Plugin	sObject data is not provided through file or row for bulk operation	File is not provided for bulk operation or in case of ROW format the sObject element is not configured
TIBCO-BW-PALETTE-SFBULK-100018	errorRole	BW-Plugin	I/O exception occurred during processing	Check files, WSDLs provided
TIBCO-BW-PALETTE-SFBULK-100019	errorRole	BW-Plugin	Response received from server is null	Not able to get a response from server. Check network connection and manually ping the Salesforce REST APIs.
TIBCO-BW-PALETTE-SFBULK-100020	errorRole	BW-Plugin	Internal error during processing	None

Error Code	Role	Category	Description	Resolution
TIBCO-BW-PALETTE-SFBULK-100021	errorRole	BW-Plugin	Job ID is not provided	Provide job ID
TIBCO-BW-PALETTE-SFBULK-100022	errorRole	BW-Plugin	Output file and format provided does not match	The output file provided to get result does not match the output format selected. Make the format and file format consistent
TIBCO-BW-PALETTE-SFBULK-100023	errorRole	BW-Plugin	Error creating get result request for Get Result activity	None
TIBCO-BW-PALETTE-SFBULK-100024	errorRole	BW-Plugin	Format mismatch between format provided in Get Result and Bulk Operation	Provide same format in Get Result as was provided for Bulk Operation
TIBCO-BW-PALETTE-SFBULK-100025	errorRole	BW-Plugin	Get result ended in error	Check on Salesforce server for failure of get result or invalid batches
TIBCO-BW-PALETTE-SFBULK-100026	errorRole	BW-Plugin	Output format provided in Get result is incorrect	Format should be either ROW, JSON, CSV, XML. Check format for Get Result.
TIBCO-BW-PALETTE-SFBULK-100027	errorRole	BW-Plugin	Error occurred for Check Status Activity	Check status failed on Salesforce. Check Salesforce server for the reason.
TIBCO-BW-PALETTE-SFBULK-100028	errorRole	BW-Plugin	Input provided for timeout or interval has incorrect format, number required	Check if timeout or interval is incorrect or is not a number
TIBCO-BW-PALETTE-SFBULK-100029	errorRole	BW-Plugin	Error creating check status request	None
TIBCO-BW-PALETTE-SFBULK-100030	errorRole	BW-Plugin	Interval not provided for Check Status	Provide interval if **Wait for Completion** is selected
TIBCO-BW-PALETTE-SFBULK-100031	errorRole	BW-Plugin	Timeout not provided for Check Status	Provide timeout value if **Wait for Completion** is selected

Error Code	Role	Category	Description	Resolution
TIBCO-BW-PALETTE-SFBULK-100032	errorRole	BW-Plugin	Query not provided for Bulk Query	Provide query string for bulk query operation

Salesforce Bulk API 2.0 Palette

Error Code	Role	Category	Description	Resolution
TIBCO-BW-PALETTE-Sfbulk-200000	errorRole	BW-Plugin	Object Mismatch: The object provided and substituted do not match in Bulk Operation	Check the object provided through the **General** or the **Input** tab and the one substituted in sObject.
TIBCO-BW-PALETTE-Sfbulk-200001	errorRole	BW-Plugin	Row Input not provided	Row format is selected. Substitute and provide input through sObject on the **Input** tab.
TIBCO-BW-PALETTE-Sfbulk-200002	errorRole	BW-Plugin	File Input not provided	Format is CSV but CSV file is not provided. Provide correct file path.
TIBCO-BW-PALETTE-Sfbulk-200003	errorRole	BW-Plugin	File not found	Check if the file provided for the operation is not present, is corrupt, not readable, or is protected.
TIBCO-BW-PALETTE-Sfbulk-200004	errorRole	BW-Plugin	External session not provided; but external session used is mapped to true	Provide external session ID and server url.
TIBCO-BW-PALETTE-Sfbulk-200005	errorRole	BW-Plugin	Connection is not provided	Provide connection on the **General** tab
TIBCO-BW-PALETTE-Sfbulk-200006	errorRole	BW-Plugin	Log in failed	Test Connection. Check username, password, user validity.
TIBCO-BW-PALETTE-Sfbulk-200007	errorRole	BW-Plugin	IO error	Error in internal processing. Check BW process and validations, engine settings.
TIBCO-BW-PALETTE-Sfbulk-200008	errorRole	BW-Plugin	Incomplete input	Check if all input required for operation is provided on the **General** or the **Input** tab.

Error Code	Role	Category	Description	Resolution
TIBCO-BW-PALETTE-Sfbulk-200009	errorRole	BW-Plugin	Error in parsing file/data	Check the file or data provided for correct the CSV format, delimiter, inconsistent data.
TIBCO-BW-PALETTE-Sfbulk-200010	errorRole	BW-Plugin	Error in firing request to Salesforce API	Check the cause of error and proceed accordingly.
TIBCO-BW-PALETTE-Sfbulk-200011	errorRole	BW-Plugin	Error in parsing JSON response	Check the validity of JSON response from Salesforce.
TIBCO-BW-PALETTE-Sfbulk-200012	errorRole	BW-Plugin	File data empty in Bulk Operation	Check if the file has records and a CSV Header. File should have CSV header and at least 1 record or some data.
TIBCO-BW-PALETTE-Sfbulk-200013	errorRole	BW-Plugin	Interval not provided for Check status	Provide positive non-zero integer value for interval in check status
TIBCO-BW-PALETTE-Sfbulk-200014	errorRole	BW-Plugin	Invalid operation provided	Check the operation provided on the **General** or the **Input** tab. It should be insert, upsert, update, delete, hardDelete, query, queryAll only.
TIBCO-BW-PALETTE-Sfbulk-200015	errorRole	BW-Plugin	Job ID is not provided	Provide job ID in the input section for GetResult or CheckStatus
TIBCO-BW-PALETTE-Sfbulk-2000016	errorRole	BW-Plugin	State is not provided	Provide the state in CloseJob
TIBCO-BW-PALETTE-Sfbulk-200017	errorRole	BW-Plugin	Empty response from Salesforce API	Check if the job is complete on the Salesforce side. Put a CheckStatus with waitForCompletion before GetResult with job appropriate timeout.
TIBCO-BW-PALETTE-Sfbulk-2000018	errorRole	BW-Plugin	Invalid result type in GetResult	Provide correct result type in GetResult. It should be Success, Failure,Unprocessed.

Error Code	Role	Category	Description	Resolution
TIBCO-BW-PALETTE-Sfbulk-200019	errorRole	BW-Plugin	File zero records	Check if the file has records and a CSV Header. File should have CSV header and at least 1 record or some data.

Salesforce Composite Palette

Error Code	Role	Category	Description	Resolution
BW-SFCOMP-20001	error	BW-Plugin	External session details not provided or incorrect details provided	Verify the external session details
BW-SFCOMP-20002	error	BW-Plugin	Error during parsing the composite tree request, required fields such as referenceId not provided	Verify that all required fields such as referenceId are provided
BW-SFCOMP-20003	error	BW-Plugin	Failed to retry session because of an error that does not let you retry or an external session provided	Check the log in details provided
BW-SFCOMP-20004	error	BW-Plugin	Failure during sending request to the Salesforce server	None
BW-SFCOMP-20005	error	BW-Plugin	Body given in Composite Batch or Composite Dependent activity is empty or cannot be parsed	Check JSON input or mappings provided
BW-SFCOMP-20006	error	BW-Plugin	Request or response has incorrect JSON format	Check JSON input value provided
BW-SFCOMP-20007	error	BW-Plugin	Internal error during processing	None
BW-SFCOMP-20008	error	BW-Plugin	Error during logging in to a Salesforce instance	Check log in details provided in the Salesforce Connection

TIBCO Documentation and Support Services

For information about this product, you can read the documentation, contact TIBCO Support, and join the TIBCO Community.

How to Access TIBCO Documentation

Documentation for TIBCO products is available on the TIBCO Product Documentation website, mainly in HTML and PDF formats.

The website is updated frequently and is more current than any other documentation included with the product.

Product-Specific Documentation

The following documentation for this product is available on the TIBCO ActiveMatrix BusinessWorks™ Plug-in for Salesforce.com product documentation page.

- *TIBCO ActiveMatrix BusinessWorks™ Plug-in for Salesforce.com Release Notes*
- *TIBCO ActiveMatrix BusinessWorks™ Plug-in for Salesforce.com Installation*
- *TIBCO ActiveMatrix BusinessWorks™ Plug-in for Salesforce.com User's Guide*

How to Contact TIBCO Support

Get an overview of TIBCO Support. You can contact TIBCO Support in the following ways:

- For accessing the Support Knowledge Base and getting personalized content about products you are interested in, visit the TIBCO Support website.

- For creating a Support case, you must have a valid maintenance or support contract with TIBCO. You also need a user name and password to log in to TIBCO Support website. If you do not have a user name, you can request one by clicking **Register** on the website.

How to Join TIBCO Community

TIBCO Community is the official channel for TIBCO customers, partners, and employee subject matter experts to share and access their collective experience. TIBCO Community offers access to Q&A forums, product wikis, and best practices. It also offers access to extensions, adapters, solution accelerators, and tools that extend and enable customers to gain full value from TIBCO products. In addition, users can submit and vote on feature requests from within the TIBCO Ideas Portal. For a free registration, go to TIBCO Community.

Legal and Third-Party Notices