Oracle® Cloud

Using the Salesforce Adapter

Release 16.4

E66628-04

December 2016

This guide describes how to configure and add the Salesforce Adapter to an integration in Oracle Integration Cloud Service.

ORACLE

Oracle Cloud Using the Salesforce Adapter, Release 16.4

E66628-04

Primary Author: Mark Kennedy

Contents

Preface

Using the Salesforce Adapter describes how to use the Salesforce Adapter as a connection in your integration.

Topics:

Audience

Using the Salesforce Adapter is intended for users who want to use the Salesforce Adapter in integrations.

Related Resources

For more information, see these Oracle resources:

- Oracle Cloud

  `http://cloud.oracle.com`
- *Using Oracle Integration Cloud Service*
- *Using the Oracle Mapper*
- *Getting Started with Oracle Cloud*
- *Managing and Monitoring Oracle Cloud*

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.

Convention	Meaning
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Getting Started with the Salesforce Adapter

Review the following conceptual topics to learn about the Salesforce Adapter and how to use it as a connection in integrations in Oracle Integration Cloud Service. A typical workflow of adapter and integration tasks is also provided.

Topics

Salesforce Adapter Capabilities

The Salesforce Adapter enables you to create an integration with Salesforce CRM applications.

The Salesforce Adapter provides the following benefits:

- Integrates easily with the Salesforce application's WSDL file to produce a simplified, integration-centric WSDL.
- Contacts the Salesforce application to fetch metadata information about business objects.
- Provides invoke (target) support for performing the following type of operations against business objects fetched from the Salesforce application:
  - CRUD (create, get, update, and destroy) operations
  - Salesforce Object Query Language (SOQL) or Salesforce Object Search Language (SOSL) query operations
- Provides invoke (outbound) support for using a custom WSDL that includes custom Apex classes exposed as SOAP web services.
- Provides trigger (source) messaging support for objects through use of the Salesforce outbound messaging WSDL.
- Provides trigger (source) callback support.

- Provides metadata caching support. For information, see Refreshing Integration Metadata.

Video

Video

Video

What Application Version Does the Salesforce Adapter Support?

The Salesforce Adapter is compatible with all supported versions of Salesforce.com. Review the end-of-life and support policy of Salesforce.com for more information.

About Oracle Integration Cloud Service

Oracle Integration Cloud Service is a complete, secure, but lightweight integration solution that enables you to connect your applications in the cloud. It simplifies connectivity between your applications and connects both your applications that live in the cloud and your applications that still live on premises. Oracle Integration Cloud Service provides secure, enterprise-grade connectivity regardless of the applications you are connecting or where they reside.

Oracle Integration Cloud Service provides native connectivity to Oracle Software as a Service (SaaS) applications, such as Oracle Sales Cloud, Oracle RightNow Cloud, and so on. Oracle Integration Cloud Service *adapters* simplify connectivity by handling the underlying complexities of connecting to applications using industry-wide best practices. You only need to create a *connection* that provides minimal connectivity information for each system. Oracle Integration Cloud Service *lookups* map the different codes or terms used by the applications you are integrating to describe similar items (such as country or gender codes). Finally, the visual data mapper enables you to quickly create direct mappings between the trigger and invoke data structures. From the mapper, you can also access lookup tables and use standard XPath functions to map data between your applications.

Once you integrate your applications and activate the integrations to the runtime environment, the dashboard displays information about the running integrations so you can monitor the status and processing statistics for each integration. The dashboard measures and tracks the performance of your transactions by capturing and reporting key information, such as throughput, the number of messages processed successfully, and the number of messages that failed processing. You can also manage business identifiers that track fields in messages and manage errors by integrations, connections, or specific integration instances.

About Oracle Integration Cloud Service Connections

Connections define information about the instances of each configuration you are integrating. Oracle Integration Cloud Service includes a set of predefined *adapters*, which are the types of applications on which you can base your connections, such as Oracle Sales Cloud, Oracle Eloqua Cloud, Oracle RightNow Cloud, and others. A connection is based on an adapter. A connection includes the additional information required by the adapter to communicate with a specific instance of an application (this can be referred to as metadata or as connection details). For example, to create a connection to a specific RightNow Cloud application instance, you must select the Oracle RightNow adapter and then specify the WSDL URL, security policy, and security credentials to connect to it.

Video

About Oracle Integration Cloud Service Integrations

Integrations are the main ingredient of Oracle Integration Cloud Service. An integration includes at the least a trigger (source) connection (for requests sent to Oracle Integration Cloud Service) and invoke (target) connection (for requests sent from Oracle Integration Cloud Service to the target) and the field mapping between those two connections.

When you create your integrations, you build on the connections you already created by defining how to process the data for the trigger (source) and invoke (target) connections. This can include defining the type of operations to perform on the data, the business objects and fields against which to perform those operations, required schemas, and so on. To make this easier, the most complex configuration tasks are handled by Oracle Integration Cloud Service. Once your trigger (source) and invoke (target) connections are configured, the mappers between the two are enabled so you can define how the information is transferred between the trigger (source) and invoke (target) data structures for both the request and response messages.

Video

About Salesforce Adapter Use Cases

The Salesforce Adapter can integrate with Salesforce.com. Use the Salesforce Adapter to send data to Salesforce.com and also receive events from Salesforce.com.

The following are some common use cases:

- Account and contact synchronization between Salesforce and ERP (such as Oracle E-Business Suite and Netsuite).
- Opportunity to order synchronization between Salesforce and ERP systems.
- Employee on-boarding from HCM systems to Salesforce.com.

Typical Workflow for Creating and Including an Adapter Connection in an Integration

You follow a very simple workflow to create a connection with an adapter and include the connection in an integration in Oracle Integration Cloud Service.

This table lists the workflow steps for both adapter tasks and overall integration tasks, and provides links to instructions for each step.

Step	Description	More Information
1	Create the adapter connections for the applications you want to integrate. The connections can be reused in multiple integrations and are typically created by the administrator.	Creating a Salesforce Adapter Connection
2	Create the integration. When you do this, you add trigger and invoke connections to the integration.	Creating an Integration and Adding the Salesforce Adapter Connection to an Integration

Step	Description	More Information
3	Map data between the trigger connection data structure and the invoke connection data structure.	Mapping Integration Cloud Service Data of *Using Oracle Integration Cloud Service*
4	(Optional) Create lookups that map the different values used by those applications to identify the same type of object (such as gender codes or country codes).	Creating Lookups of *Using Oracle Integration Cloud Service*
5	Activate the integration.	Managing Integrations of *Using Oracle Integration Cloud Service*
6	Monitor the integration on the dashboard.	Monitoring Integration Cloud Services of *Using Oracle Integration Cloud Service*
7	Track payload fields in messages during runtime.	Assigning Business Identifiers for Tracking Fields in Messages and Managing Business Identifiers for Tracking Fields in Messages of *UsingOracle Integration Cloud Service*
8	Manage errors at the integration level, connection level, or specific integration instance level.	Managing Errors of *Using Oracle Integration Cloud Service*

2

Creating a Salesforce Adapter Connection

A connection is based on an adapter. You define connections to the specific cloud applications that you want to integrate. The following topics describe how to define connections.

Topics

Prerequisites for Creating a Connection

You must satisfy the following prerequisites to create a connection with the Salesforce Adapter:

- Understand Salesforce constraints. For more information, see Understanding Salesforce Cloud Constraints.
- If you are new to Salesforce, you must create a free Salesforce development organization. This registration provides you with the ability to create a WSDL to use in an integration. For information, visit https://www.salesforce.com.
- Create the type of WSDL to use in an integration:
  - Enterprise WSDL: For information, see Creating the Salesforce Adapter Enterprise WSDL.
  - Outbound messaging WSDL: For information, see Creating the Salesforce Adapter Outbound Messaging WSDL.
  - Custom WSDL: For information, see Creating the Salesforce Adapter Custom WSDL.

Understanding Salesforce Cloud Constraints

You must be aware of the following constraints before configuring the Salesforce Adapter.

- The Salesforce Adapter uses the SalesForceDotCom (SFDC) API for all activities. Therefore, it is subject to any Salesforce API limitations. The limitations are defined in the Salesforce Limits Quick Reference Guide.
- Not all the push topic queries are supported by Salesforce. See Supported Push Topic Queries and Unsupported PushTopic Queries.
- Client applications must adhere to Salesforce's SOAP API support policy and backward compatibility terms. These terms are available at SFDC SOAP API Support Policy.

Creating the Salesforce Adapter Enterprise WSDL

You must create the Salesforce Adapter enterprise WSDL to include in an integration. You then specify this WSDL when creating a Salesforce Adapter connection on the Connections page.

To create the Salesforce Adapter enterprise WSDL:

1. Log in to your Enterprise, Unlimited, or Developer Edition Salesforce.com account. Open the Web browser and enter the following URL:

   `www.salesforce.com`

2. Log in to `Salesforce.com` using a valid user name and password.

   You must log in as an administrator or user with the **Modify All Data** permission. Logins are checked to ensure that they are from a known IP address.

3. Under **App Setup**, Expand **Develop** and click **API** to display the WSDL download page.

4. If the organization has managed packages installed in the organization, click **Generate Enterprise WSDL**. Salesforce prompts you to select the version of each installed package to include in the generated WSDL or right-click **Generate Enterprise WSDL** and save it to a local directory.

   In the right-click menu, Internet Explorer users can choose **Save Target As**, while Mozilla Firefox users can choose **Save Link As** to save it to the local directory.

   The Save dialog is displayed.

5. Provide a name for the WSDL file and a location to save the file on your file system, and click **Save**. For information about uploading this WSDL when creating a connection, see Configuring Connection Properties.

Creating the Salesforce Adapter Outbound Messaging WSDL

You can create an outbound messaging WSDL for the Salesforce Adapter. You then select this WSDL when configuring the Salesforce Adapter as a trigger in the Outbound Messaging page of the Adapter Endpoint Configuration Wizard.

This process consists of several steps:

- The outbound message consists of a workflow, approval, or milestone action that sends your specified information to your specified endpoint. You configure outbound messaging in the Salesforce setup menu. Afterward, you configure the endpoint.

  To create a workflow rule:

1. Log in to your Salesforce account and go to **Setup**.
2. Under the **App Setup** menu, expand **Create**, followed by **Workflow & Approvals**.
3. Select a workflow rule or approval process as per your integration requirement.
4. Click **Create New**, provide the required information in the following wizards, and click **Save**.
    a. For the workflow rule, click **Edit** under the **Workflow Action** menu followed by **Add Workflow Action**, and then **New Outbound Message**.
    b. For the approval process, click **Add New** (you can select for one or more actions including **Submission**, **Approval**, **Rejection**, and **Recall**) followed by **New Outbound Message**.

Outbound messaging WSDLs associated with approval processes or entitlement processes are also supported and consumed by the adapter.

- Create the Salesforce outbound messaging WSDL at www.salesforce.com. You then select this WSDL to receive outbound message notifications from the Salesforce application on the Outbound Messaging page in the Adapter Endpoint Configuration Wizard. For instructions, see What You See on the Salesforce Cloud Trigger Outbound Messaging Properties.

  To create the Salesforce outbound messaging WSDL:

  1. Log in to your Salesforce account and go to **Setup** > **Outbound Messages**.
  2. Select the required object, and click **Next**.
  3. Enter other required details (in the **Endpoint URL** field, enter a dummy URL), and click **Save**.
  4. Click **Generate WSDL** to download the WSDL.
  5. Drag the Salesforce Adapter to the trigger (inbound) section of the integration canvas. This invokes the Adapter Endpoint Configuration Wizard.
  6. Browse for the generated WSDL on the Outbound Messaging page.
  7. Activate the integration and copy the endpoint URL from the integration information icon.
  8. Go to the **Outbound Messaging** section at www.salesforce.com and replace the dummy URL you entered in Step 3 with the real endpoint URL.

Creating the Salesforce Adapter Custom WSDL

You can create a custom WSDL that includes custom Apex classes written on `force.com` and exposed as SOAP web services. This enables external applications to access your code and application.

You then select the custom WSDL when configuring the Salesforce Adapter in the invoke direction on the Basic Info page of the Adapter Endpoint Configuration Wizard.. For instructions, see What You See on the Salesforce Cloud Invoke Basic Info Page.

For more information about custom WSDLs, see Exposing Apex Methods as SOAP Web Services.

Uploading an SSL Certificate

Certificates are used to validate outbound SSL connections. If you make an SSL connection in which the root certificate does not exist in Oracle Integration Cloud Service, an exception is thrown. In that case, you must upload the appropriate certificate. A certificate enables Oracle Integration Cloud Service to connect with external services. If the external endpoint requires a specific certificate, request the certificate and then upload it into Oracle Integration Cloud Service.

To upload a certificate:

1. From the Oracle Integration Cloud Service home page, click the **Administration** tab in the upper right corner.

   All certificates currently uploaded to the trust store are displayed in the Certificates dialog. The **Filter By** > **Type** list displays the following details:

   - **Preinstalled**: Displays the certificates automatically installed in Oracle Integration Cloud Service. These certificates cannot be deleted.
   - **Uploaded**: Displays the certificates uploaded by individual users. These certificates can be deleted and updated.

   You can also search for certificates in the **Search** field. The search results are limited to a maximum of ten records sorted by name for performance and usability reasons. To ensure that your search results are more granular, enter as much of the certificate name as possible.

2. Click **Upload** at the top of the page.
3. In the Upload Certificate dialog box, enter a unique identifier for the certificate.

   This is a name you can use to identify the certificate.

4. Click **Browse** to locate the certificate file (`.cer`).
5. Click **Upload**.
6. Click the certificate name to view details such as the subject of the certificate, the issuer of the certificate, the date the certificate was issued, and the date the certificate expires.

Creating a Connection

The first step in creating an integration is to create the connections to the applications with which you want to share data.

1. In the Integration Cloud Service toolbar, click **Designer**.
2. On the Designer Portal, click **Connections**.
3. Click **New Connection**.

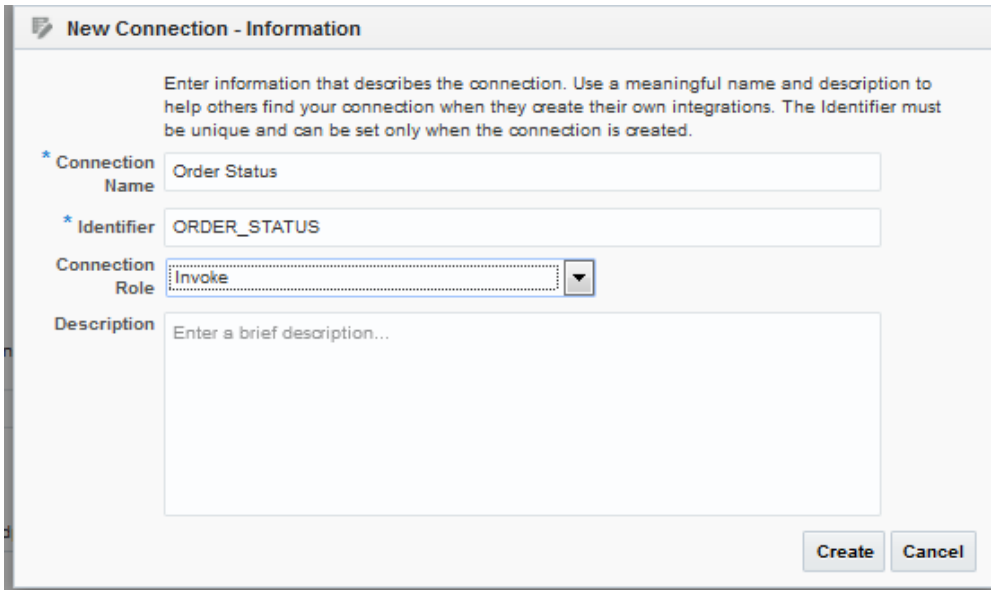
   The Create Connection — Select Adapter dialog is displayed.

4. Select an adapter from the dialog. You can also search for the type of adapter to use by entering a partial or full name in the Search field, and clicking **Search**.

The New Connection — Information dialog is displayed.

5. Enter the information to describe the connection.

   - Enter a meaningful name to help others find your connection when they begin to create their own integrations. The name you enter is automatically added in capital letters to the **Identifier** field. If you modify the identifier name, do not include a blank space (for example, `OSC Inbound`).
   - Select the role (direction) in which to use this connection (trigger, invoke, or both). Only the roles supported by this adapter are displayed for selection. When you select a role, only the connection properties and security policies appropriate to that role are displayed on the Connections page. If you select an adapter that supports both invoke and trigger, but select only one of those roles, then try to drag the adapter into the section you did not select, you receive an error (for example, configure an Oracle RightNow Cloud Adapter as only an invoke, but drag the adapter to the trigger section).
   - Enter an optional description of the connection.



6. Click **Create**.

   Your connection is created and you are now ready to configure connection details, such as email contact, connection properties, security policies, and connection login credentials.

Adding a Contact Email

From the Connection Administrator section of the connection, you can add a contact email address for notifications.

1. In the **Email Address** field, enter an email address to receive email notifications when problems occur.
2. In the upper right corner, click **Save**.

Configuring Connection Properties

Enter connection information so your application can process requests.

1. Click **Configure Connectivity**.

   The Connection Properties dialog is displayed.

2. Select the **Upload File** checkbox, then click **Upload** to select the enterprise WSDL URL to use in this integration. The enterprise WSDL is mandatory and must be specified regardless of whether you are also using a custom WSDL or an outbound messaging WSDL. For instructions on creating this WSDL, see Creating the Salesforce Adapter Enterprise WSDL.

3. Click **OK**.

4. Configure connection security. See Configuring Connection Security.

Configuring Connection Security

Configure security for your Salesforce Adapter connection by selecting the security policy and security token.

1. Click **Configure Credentials**.

2. Enter your login credentials:

   a. Select the security policy. Only the Salesforce Login Token policy is supported. It cannot be deselected.

   b. Enter the username and password that provide access to the destination web service.

   c. Reenter the password a second time.

3. Click **OK**.

Testing the Connection

Test your connection to ensure that it is successfully configured.

1. In the upper right corner of the page, click **Test**.

   If successful, the following message is displayed and the progress indicator shows 100%.

   ```
   The connection test was successful!
   ```

2. If your connection was unsuccessful, an error message is displayed with details. Verify that the configuration details you entered are correct.

3. When complete, click **Save**.

Editing a Connection

You can edit connection settings after creating a new connection.

1. In the Oracle Integration Cloud Service toolbar, click **Designer**.

2. On the Designer Portal, click **Connections**.
3. On the Connections page, search for the connection name.
4. Select **Edit** from the connection **Actions** menu or click the connection name.

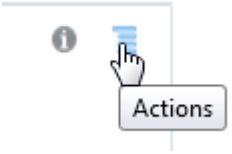
The Connection page is displayed.

5. To edit the notification email contact, change the email address in the **Email Address** field.
6. To edit the connection properties, click **Configure Connectivity**. Note that some connections do not include this button. If your connector does not include a **Configure Connectivity** button, then click the **Configure Credentials** button.

Cloning a Connection

You can clone a copy of an existing connection. It is a quick way to create a new connection.

1. In the Oracle Integration Cloud Service toolbar, click **Designer**.
2. On the Designer Portal, click **Connections**.
3. On the Connections page, search for the connection name.
4. Select **Clone** from the connection **Actions** menu.



The Clone Connection dialog is displayed.

5. Enter the connection information.
6. Click **Clone**.
7. Click **Edit** to configure the credentials of your cloned connection. Cloning a connection does not copy the credentials.

See Editing a Connection for instructions.

Deleting a Connection

You can delete a connection from the connection menu.

1. In the Oracle Integration Cloud Service toolbar, click **Designer**.

2. On the Designer Portal, click **Connections**.
3. On the Connections page, search for the connection name.
4. Click **Delete** from the connection **Actions** menu.

The Delete Connection dialog is displayed if the connection is not used in an integration.

5. Click **Yes** to confirm deletion.

Refreshing Integration Metadata

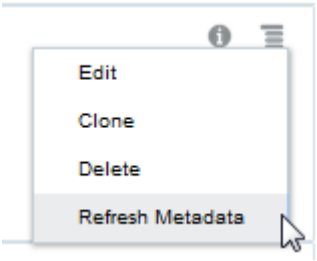
You can manually refresh the currently-cached metadata available to adapters that have implemented metadata caching. Metadata changes typically relate to customizations of integrations, such as adding custom objects and attributes to integrations. There may also be cases in which integrations have been patched, which results in additional custom objects and attributes being added. This option is similar to clearing the cache in your browser. Without a manual refresh, a staleness check is only performed when you drag a connection into an integration. This is typically sufficient, but in some cases you may know that a refresh is required. For these cases, the **Refresh Metadata** menu option is provided.

To refresh integration metadata:

Note: The **Refresh Metadata** menu option is only available with adapters that have implemented metadata caching.

---

1. In the Integration Cloud Service toolbar, click **Designer**.
2. In the Designer Portal, click **Connections**.
3. Locate the connection to refresh.
4. From the menu at the right, select **Refresh Metadata**.



A message is displayed indicating that the refresh was successful.

```
Metadata refresh for connection "connection_type" has been initiated successfully.
```

3

Creating an Integration

Integrations use the adapter connections you created to your applications, and define how information is shared between those applications. You can view, export, create, import, edit, or delete integrations; create integrations to publish or subscribe to messages; add and remove request and response enrichment triggers; and create routing paths for different target endpoints in integrations. Click the following topics for more information.

Topic

- Creating Integrations (in *Using Oracle Integration Cloud Service*)

4

Adding the Salesforce Adapter Connection to an Integration

When you drag the Salesforce Adapter into the trigger and invoke areas of an integration, the Adapter Endpoint Configuration Wizard is invoked. This wizard guides you through configuration of the Salesforce Adapterendpoint properties.

The following sections describe the wizard pages that guide you through configuration of the Salesforce Adapter as a trigger or invoke in an integration.

Topics

For more information about the Salesforce Adapter, see Salesforce Adapter Capabilities.

Configuring Basic Information Properties

You can enter a name and description on the Basic Info page of each adapter in your integration.

Topics

What You Can Do from the Basic Info Page

You can specify the following values on the Basic Info page. The Basic Info page is the initial wizard page that is displayed whenever you drag an adapter to the section of the integration canvas supported by your adapter.

- Specify a meaningful name.
- Specify a description of the responsibilities.

What You See on the Basic Info Page

The following table describes the key information on the Basic Info page.

Element	Description
What do you want to call your endpoint?	Provide a meaningful name so that others can understand the responsibilities of this connection. You can include English alphabetic characters, numbers, underscores, and dashes in the name. You cannot include the following: • Blank spaces (for example, `My Inbound Connection`) • Special characters (for example, `#;83&` or `righ(t)now4`) • Multibyte characters
What does this endpoint do?	Enter an optional description of the connection's responsibilities. For example: `This connection receives an inbound request to synchronize account information with the cloud application.`

Configuring Salesforce Trigger Outbound Messaging Properties

Enter the Salesforce trigger outbound messaging values for your integration.

Topics

What You Can Do from the Salesforce Cloud Trigger Outbound Messaging Properties Page

You can select the trigger outbound messaging WSDL to use with the Salesforce Adapter.

You must have already created this WSDL. This process consists of several steps. For information on creating the outbound Messaging WSDL, see Creating the Salesforce Adapter Outbound Messaging WSDL.

What You See on the Salesforce Cloud Trigger Outbound Messaging Properties

The following table describes the key information on the trigger Outbound Messaging page.

Element	Description
Select the Outbound Messaging WSDL	Select the invoke Salesforce outbound messaging WSDL to receive outbound message notifications from the Salesforce application. **Note**: You must first create a workflow rule and generate an outbound messaging WSDL as described in Creating the Salesforce Adapter Outbound Messaging WSDL. Outbound messaging WSDLs associated with approval processes or entitlement processes are also supported and consumed by the adapter. 1. Browse for and select the invoke Salesforce outbound messaging WSDL 2. Activate the integration and copy the endpoint URL from the integration information icon. 3. Go to the **Outbound Messaging** section at www.salesforce.com and replace the dummy URL you entered in Creating the Salesforce Adapter Outbound Messaging WSDL with the real endpoint URL

Configuring Salesforce Trigger Response Properties

Enter the Salesforce trigger response values for your integration.

Topics

- What You Can Do from the Salesforce Trigger Response Page
- What You See on the Salesforce Cloud Trigger Response Page

What You Can Do from the Salesforce Trigger Response Page

You can configure the response parameters for the Salesforce Adapter.

You can select the type of callback response to send as a response document from the integration flow to the Salesforce Cloud application.

- Configure to send no callback response.
- Configure the operation and business objects to use for a successful callback response.
- Configure the operation and business objects for a callback response for a failed integration flow.

What You See on the Salesforce Cloud Trigger Response Page

The following table describes the key information on the trigger Response page.

Element	Description
Send a response	Deselect if no callback response is required.
Configuration a Successful Callback Response	Select **Configure** to configure the operation and business objects to use for a successful callback response.
Configuration a Failure Callback Response	Select **Configure** to configure the operation and business objects for a callback response for a failed integration flow. The option to configure a failure callback response gets enabled only after the configuration of successful callback response.
Edit	Click to edit the operation or business objects of a successful or failed callback response. This button is displayed after you configure a successful callback response, a failed callback response, or both.
Header	Click to configure the header properties for the selected operation. The headers available for configuration are based on the type of operation you selected. This button is displayed after you configure a successful callback response, a failed callback response, or both.
Reset	Click to reset the operation and header configuration to the default values. This button is displayed after you configure a successful callback response, a failed callback response, or both.

Configuring Salesforce Trigger Callback Response Properties

Enter the Salesforce trigger callback response values for your integration.

Topics

What You Can Do from the Salesforce Trigger Callback Response Page

You can configure the callback response parameters for the Salesforce Adapter.

- Configure the operation and business objects to use for a successful callback response.
- Configure the operation and business objects for a callback response for a failed integration flow.

What You See on the Salesforce Cloud Trigger Callback Response Page

The following table describes the key information on the trigger Callback Response page.

Element	Description
Select an Operation Type	Select the type of operation to perform on the business objects in a Salesforce Cloud application: • **CORE**: Displays the following selections: **ConvertLead**, **Merge**, **Undelete**, or **Upsert**. • **CRUD**: Represents the create, read, update, delete, or destroy operations to perform on Salesforce Cloud business objects. Each letter maps to a standard SQL statement, HTTP method, or DDS operation. Select the CRUD operation to perform on the business object: **Create**, **Delete**, or **Update**. **Note**: For the **Update** operation, external ID cannot be mapped to ID. • **MISC**: Represents the set of specialized task operations to perform in the Salesforce Cloud application.
Filter by object name	Enter the initial letters of an object name to display a range of objects. You can also enter an asterisk (*) after the query in the search field (for example, to search for all objects starting with `Acc`, enter `Acc*`). You can also select a filter type: • **All**: Displays all objects. • **Custom**: Displays objects you created. The naming convention is a combination of the object name appended with `_c`. • **Standard**: Displays business objects delivered as part of the Salesforce Cloud application.

Element	Description
Select Business Objects (Salesforce API *version*)	Select a single business object or multiple business objects from the Salesforce Cloud application. The selected operation acts upon these business objects. When you complete invoke operation configuration, the selected operation and business objects are defined in the integration-centric WSDL file.
Your Selected Business Objects	Displays the business objects you selected.

Configuring Salesforce Invoke Basic Information Properties

You can enter a name and description and select the type of WSDL to use on the Salesforce invoke Basic Info page.

Topics

What You Can Do from the Salesforce Cloud Invoke Basic Info Page

You can specify the following values on the Salesforce Cloud target Basic Info page. The Salesforce Cloud invoke Basic Info page is the initial wizard page that is displayed whenever you drag an adapter to the invoke area.

- Specify a meaningful name.
- Specify a description of the connection responsibilities.
- Select to use a standard application delivered by `salesforce.com` (enterprise WSDL) or custom application built using Apex classes and hosted on `force.com` (custom WSDL).

What You See on the Salesforce Cloud Invoke Basic Info Page

The following table describes the key information on the Salesforce Cloud invoke Basic Info page.

Element	Description
What do you want to call your endpoint?	Provide a meaningful name so that others can understand the connection. For example, if you are creating an invoke Salesforce Cloud connection, you may want to name it `SalesforceOutboundDirection`. You can include English alphabetic characters, numbers, underscores, and dashes in the name. You cannot include the following: • Blank spaces (for example, `My Salesforce Connection`) • Special characters (for example, `#;83&` or `righ(t)now4`) • Multibyte characters
What does this endpoint do?	Enter an optional description of the connection's responsibilities. For example: `This connection receives an outbound request to synchronize account information with the Salesforce Cloud Application.`
Select outbound support option	Select the type of WSDL to use. • **Standard applications delivered by Salesforce.com**: Select this option if you want to use the enterprise WSDL that you specified in the Connection Properties dialog during adapter configuration. If you select this option, you are taken to the Operations page to select an operation type and business object to use. • **Custom applications built using Apex Classes and hosted on force.com**: Select this option if you want to use a custom WSDL that includes custom Apex classes written on `force.com` and exposed as SOAP web services. This enables external applications to access your code and application. If you select this option, you are taken to the Custom Operations page. For more information about creating custom WSDLs, see Creating the Salesforce Adapter Custom WSDL.

Configuring Salesforce Invoke Operation Properties

Enter the Salesforce invoke operation values for your integration.

Topics

- What You See on the Salesforce Target Operation Page

What You Can Do from the Salesforce Cloud Invoke Operations Page

You can configure the following invoke operations values for Salesforce Cloud.

- Select either of the following operation types:
  1. CRUD
  2. Salesforce Object Query Language (SOQL) or Salesforce Object Search Language (SOSL) query
- Select the business objects.
- Specify the SOQL/SOSL query.

What You See on the Salesforce Invoke Operations Page

The following table describes the key information on the Salesforce Cloud invoke Operations page.

Element	Description

Select an Operation Type	Select the type of operation to perform:
	• **CORE** : Represents all core operations supported by the Salesforce application.
	• **CRUD**: Represents the create, read, update, delete, or destroy operation to perform on Salesforce business objects. Each letter maps to a standard SQL statement, HTTP method, or DDS operation. Select the CRUD operation to perform: **Create**, **Delete**, **Retrieve**, or **Update**.
	Note: For the **Update** operation, external ID cannot be mapped to ID.
	• **MISC**: Represents specialized task operations (such as fetching user information associated with the current session) in the Salesforce application.
	• **SOSL/SOQL**: Select to enter a Salesforce Object Query Language (SOQL) or Salesforce Object Search Language (SOSL) query to send as a request to the Salesforce application. The following operations are available:
	– **query**: Executes a query against specific criteria and returns data matching that criteria. Only records not deleted from your Salesforce application account are returned.
	– **queryAll**: Returns the same data as the **query** operation, along with deleted records present in the recycle bin.
	– **search**: Returns records from the Salesforce application. You can specify binding parameters to dynamically provide a search string as input to your search operation.
	If you select this option, the page is refreshed to display a field for entering an SOQL or SOSL query to send for validation:
	– **Query Statement**: Enter a valid query statement. SOQL statements evaluate to a list of sObjects, a single sObject, or an integer for `count` method queries. The following examples are provided:
	`"SELECT Id FROM Contact WHERE Name LIKE 'A%' AND MailingCity = 'California'"`
	`SELECT COUNT() FROM Contact`
	SOSL statements evaluate to a list of sObjects, where each list contains the search results for a particular sObject type. For example:
	`"SELECT a.name, a.id, a.accountNumber, c.name from Contact c, c.Account"`
	– **Binding Parameters**: Displays any parameters included in the query. For example, `orgId` is a parameter in the following query:
	`SELECT a.name, a.id, a.accountNumber, c.name from Contact c, c.Account a WHERE a.name = "&orgId"`
	This query displays a binding parameters text box in which to enter a test value for `orgId`.
	– **Test My Query**. Click to validate the query against the Salesforce application. Query results are displayed. If errors occur, you receive results about how to correct the query.

Element	Description
Filter By Object Name	Type the initial letters to filter the display of business objects. You can also select a filter type: • **All**: Displays all objects. • **Custom**: Displays objects you created. Custom business objects are appended with "_c." • **Standard**: Business objects delivered as part of the Salesforce application.
Select Business Objects	Select a single or multiple business objects to include in the operation. You can select up to ten objects for one operation.

Configuring Salesforce Invoke Header Properties

Enter the Salesforce invoke header values for your integration.

Topics

What You Can Do from the Salesforce Cloud Invoke Headers Page

You can configure the invoke header properties for Salesforce Cloud.

What You See on the Salesforce Invoke Headers Page

The following table describes the key information on the Salesforce Cloud invoke Headers page.

The headers available for configuration are based on the operation you selected on the invoke Operations page. There are two types of headers:

- Request headers are sent with the request message to the Salesforce application.
- Response headers are received with the response message sent from the Salesforce application.

Visit www.salesforce.com and specify the specific name of the header property in the search utility.

Element	Description
AllOrNoneHeader (request header)	Specifies the transactional behavior for Salesforce application operations. The behavior of this parameter is based on the version of the integration and is applicable to the create, delete, update, undelete, and upsert operations. • Integration flows created before version 16.4.1: – If set to **true** and the response contains error elements, it maps to `UnexpectedErrorFault`. This is also the case if **AllOrNoneHeader** is set to **false** (that is, unselected). • Integration flows created for version 16.4.1 and later: – If set to **true** (that is, selected and there are error elements), it maps to `UnexpectedErrorFault`. – If set to **false** (that is, unselected), the adapter returns the whole response even if it contains error elements along with success elements in the response. If you want integration flows created before 16.4.1 to use the new behavior, edit the Salesforce Adapter in the Adapter Endpoint Configuration Wizard.
AllowFieldTruncationHeader (request header)	Specifies the truncation behavior for the following fields (each are string data types): • `anyType` • `email` • `picklist` • `encryptedstring` • `textarea` • `mulitpicklist` • `phone` • `string` Set **allowFieldTruncation** to one of the following values: • **True**: If you enter a value of 25 characters in a field of 20 characters, the first 20 records are inserted into the field and the transaction is successful. • **False**: If you enter a value of 25 characters in a field of 20 characters, an error is thrown and the transaction does not commit.

Element	Description
AssignmentRuleHeader (request header)	Specifies the assignment rule to use when creating or updating an account, case, or lead. The assignment rule can be active or inactive. The ID is retrieved by querying the `AssignmentRule` object. If the ID is specified, you do not need to specify the **useDefaultRule** value. • **assignmentRuleId**: The ID of the assignment rule to use. The ID is not validated by the Salesforce Cloud application, whether or not it exists. Validation occurs during runtime. • **useDefaultRule**: If set to **true**, the default (active) assignment rule is used. If set to **false**, the default (active) assignment rule is not used.
EmailHeader (request header)	Specifies whether or not to send a notification email. You can set the following properties: • **triggerAutoResponseEmail** – **true**: Triggers automatic response rules for leads and cases. – **false**: Automatic response rules for leads and cases are not triggered. • **triggerOtherEmail** – **true**: The email is triggered outside the organization. – **false**: The email is not triggered outside the organization. • **triggerUserEmail** – **true**: The email is triggered and sent to users in the organization. This email is triggered by a number of events such as adding comments to a case or updating a task. – **false**: The email is not triggered and sent to users in the organization.
DebuggingHeader (request header)	Specify the debugging log level. The following log levels are supported: • **NONE** (least verbose) • **DEBUGONLY** • **DB** • **PROFILING** • **CALLOUT** • **DETAIL** (most verbose)

Element	Description
MruHeader (request header)	The Salesforce application shows the most recently used (MRU) items. In API version 7.0 or later, the list is not updated by itself. Use **MruHeader** to update the list. Using this header can negatively impact performance. Set **updateMru** to one of the following values: • **true**: The list of MRU items is updated in the Salesforce application. • **false**: The list of most recently used items is not updated in the Salesforce application.
PackageVersionHeader (request header)	Specifies the package version for any installed package. The package version identifies the components in a package. The package version follows the format `majorNumber.minorNumber.patchNumber` (for example, `3.4.5`, where `3` refers to `majorNumber`, `4` refers to `minorNumber`, and `5` refers to `patchNumber`) .
QueryOptions (request header)	Specifies the batch size for queries. The default value is `500`, the minimum value is `200`, and the maximum value is `2000`.
DebuggingInfo (response header)	This information is only returned if the **debugLevel** request header is sent with the request payload to the Salesforce application.
LimitInfoHeader (response header)	Provides information about the limitations of API calls on a per-day basis for the organization. • **current**: The number of calls already used in the organization. • **Limit**: The organization's limit for the specified limit type. • **Type**: The limit information type specified in the header `API REQUESTS` (contains limit information about API calls for the organization).

Configuring Salesforce Invoke Custom Operations Properties

Specify the following values on the Salesforce invoke Custom Operations page.

Topics

What You Can Do from the Salesforce Cloud Invoke Custom Operations Page

You can specify the following values on the Salesforce Cloud invoke Custom Operations page.

- Select the custom WSDL.
- Select the operations to perform.
- Optionally select a new custom WSDL to use.

What You See on the Salesforce Cloud Invoke Custom Operations Page

The following table describes the key information on the Salesforce Cloud invoke Custom Operations page.

Element	Description
Select the Custom WSDL	Select the custom WSDL to use.
Operations in Uploaded WSDL	Displays the list of operations included in the uploaded custom WSDL. Select the operation to perform in the Salesforce Cloud application.
Select New Custom WSDL	Click **Choose File** to select the custom WSDL to use. This selection replaces any previously uploaded WSDL file. After selecting the new WSDL, return to the **Operations in Uploaded WSDL** list and select the new operation to use. To use a custom WSDL that includes custom Apex classes written on `force.com` and exposed as SOAP web services, see Exposing Apex Methods as SOAP Web Services.

Reviewing Configuration Values on the Summary Page

You can review the specified adapter configuration values on the Summary page.

Topics

What You Can Do from the Summary Page

You can review configuration details from the Summary page. The Summary page is the final wizard page for each adapter after you have completed your configuration.

- View the configuration details you defined for the adapter. For example, if you have defined an inbound trigger (source) adapter with a request business object and immediate response business object, specific details about this configuration are displayed on the Summary page.
- Click **Done** if you want to save your configuration details.

- Click a specific tab in the left panel or click **Back** to access a specific page to update your configuration definitions.
- Click **Cancel** to cancel your configuration details.

What You See on the Summary Page

The following table describes the key information on the Summary page.

Element	Description
Summary	Displays a summary of the configuration values you defined on previous pages of the wizard. The information that is displayed can vary by adapter. For some adapters, the selected business objects and operation name are displayed. For adapters for which a generated XSD file is provided, click the XSD link to view a read-only version of the file. To return to a previous page to update any values, click the appropriate tab in the left panel or click **Back**.

Performing Salesforce Adapter Postconfiguration Tasks

After activating your integration, you must update the outbound message for the Salesforce adapter to send messages to Oracle Integration Cloud Service. This section describes how to activate a workflow rule.

1. Open the Salesforce application.
2. Scroll down and click **Workflow Rules.**
3. In the **Workflow Rules** panel, click the workflow rule.
4. Scroll down to the **Immediate Workflow Actions** section and click the outbound message.
5. In the **Outbound Message** panel, click **Edit**.
6. In the **Edit Outbound Message** panel, enter the endpoint URL from the **Integration Details** icon for the integration.
7. In the **Edit Outbound Message** panel, click **Save**.

   The **Outbound Message** panel is displayed.
8. In the **Outbound Message** panel, scroll down and find the **Workflow Rules Using This Outbound Message** section.
9. Click the workflow link.

   The **Workflow Rule** panel is displayed.
10. In the **Workflow Rule** panel, click **Activate**.

Your workflow is activated. The Salesforce application starts sending messages to the integration endpoint URL created when you activated the integration.

5

Creating Mappings and Lookups in Integrations

You must map data between trigger connections and invoke connections in integrations. You can also optionally create lookups in integrations.

Topics

- Mapping Integration Cloud Service Data (in *Using Oracle Integration Cloud Service*)
- Creating Lookups (in *Using Oracle Integration Cloud Service*)

6

Administering Integrations

Oracle Integration Cloud Service provides you with the information and tools required to activate, monitor, and manage your integrations in the runtime environment.

Topic

- Administering Integration Cloud Service (in *Using Oracle Integration Cloud Service*)