Oracle® Cloud

Using the Salesforce Adapter with Oracle Integration

E85419-18
September 2019
# Contents

## Preface

<table>
<thead>
<tr>
<th>Audience</th>
<th>v</th>
</tr>
</thead>
<tbody>
<tr>
<td>Documentation Accessibility</td>
<td>v</td>
</tr>
<tr>
<td>Related Resources</td>
<td>v</td>
</tr>
<tr>
<td>Conventions</td>
<td>vi</td>
</tr>
</tbody>
</table>

## 1 Understand the Salesforce Adapter

<table>
<thead>
<tr>
<th>Salesforce Adapter Capabilities</th>
<th>1-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>What Application Version Is Supported?</td>
<td>1-2</td>
</tr>
<tr>
<td>About Salesforce Adapter Use Cases</td>
<td>1-2</td>
</tr>
<tr>
<td>Workflow to Create and Add a Salesforce Adapter Connection to an Integration</td>
<td>1-2</td>
</tr>
</tbody>
</table>

## 2 Create a Salesforce Adapter Connection

<table>
<thead>
<tr>
<th>Prerequisites for Creating a Connection</th>
<th>2-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Understand Salesforce Constraints</td>
<td>2-1</td>
</tr>
<tr>
<td>Create the Salesforce Adapter Enterprise WSDL</td>
<td>2-2</td>
</tr>
<tr>
<td>Create the Salesforce Adapter Outbound Messaging WSDL</td>
<td>2-2</td>
</tr>
<tr>
<td>Create the Salesforce Adapter Custom WSDL</td>
<td>2-3</td>
</tr>
<tr>
<td>Use the Salesforce OAuth Three-Legged or Resource Owner Password Credentials Security Policy</td>
<td>2-3</td>
</tr>
<tr>
<td>Create a Connection</td>
<td>2-4</td>
</tr>
<tr>
<td>Add a Contact Email</td>
<td>2-6</td>
</tr>
<tr>
<td>Configure Connection Properties</td>
<td>2-6</td>
</tr>
<tr>
<td>Configure Connection Security</td>
<td>2-7</td>
</tr>
<tr>
<td>Test the Connection</td>
<td>2-8</td>
</tr>
<tr>
<td>Upload an SSL Certificate</td>
<td>2-9</td>
</tr>
<tr>
<td>Refresh Integration Metadata</td>
<td>2-10</td>
</tr>
</tbody>
</table>

## 3 Add the Salesforce Adapter Connection to an Integration

| Basic Info Page | 3-1  |

---

The contents listed above provide a structured overview of the document. Each section is further divided into subtopics, making it easier to navigate and understand the content. The document appears to be a guide on how to understand and create a Salesforce Adapter connection, followed by steps on how to add this connection to an integration. The guide also includes sections on basic information and prerequisites for creating a connection.
4 Implement Common Patterns Using the Salesforce Adapter

Process Large Data Sets Asynchronously with Different Bulk Operations 4-1
Use Bulk Response Operations in an Integration 4-5
Retrieve the Name and IDs from Account Records with the queryMore Operation 4-17
Send Multiple Records in a Single Message 4-21
Allow and Prevent Duplicate Headers with the Duplicate Rule Header Property 4-23
  Allow Duplicate Records with the Duplicate Rule Header Property 4-24
  Prevent Duplicate Records with the Duplicate Rule Header Property 4-26

5 Troubleshoot the Salesforce Adapter

Error When a Connected Application Attempts to Access Your Salesforce Data 5-1
Errors When the Salesforce Endpoint is Configured as a Trigger 5-1
Invoke Oracle Integration When a Payload is Not Expected 5-2
Id Field Identified with an Asterisk in the Request Mapper of an Upsert Operation is Not Mandatory 5-2
Custom and Inbound WSDLs Unavailable after Migration 5-2
Preface

This guide describes how to configure the Salesforce Adapter as a connection in an integration in Oracle Integration.

Note:
The information in this guide applies to all of your Oracle Integration instances. It doesn't matter which edition you're using, what features you have, or who manages your cloud environment. You'll find what you need here, including notes about any differences between the various flavors of Oracle Integration when necessary.

Topics
• Audience
• Documentation Accessibility
• Related Resources
• Conventions

Audience

This guide is intended for developers who want to use the Salesforce Adapter in integrations in Oracle Integration.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Related Resources

See these Oracle resources:
• Oracle Cloud
  http://cloud.oracle.com
• Using Integrations in Oracle Integration
• Using the Oracle Mapper with Oracle Integration

Conventions

The following text conventions are used in this document:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>boldface</strong></td>
<td>Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.</td>
</tr>
<tr>
<td><em>italic</em></td>
<td>Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.</td>
</tr>
<tr>
<td><code>monospace</code></td>
<td>Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.</td>
</tr>
</tbody>
</table>
Understand 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. A typical workflow of adapter and integration tasks is also provided.

Topics:

• Salesforce Adapter Capabilities
• What Application Version Is Supported?
• About Salesforce Adapter Use Cases
• Workflow to Create and Add a Salesforce Adapter Connection to an Integration

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:
  – Bulk (Create, Update, Upsert, Delete, FinalBatch, getStatusForAllBatches, and getBatchResults)
  – Core (Convertlead, GetDeleted, GetUpdated, Process, Merge, Undelete, and Upsert)
  – CRUD (create, delete, retrieve, and update) operations
  – Salesforce Object Query Language (SOQL) query and queryAll operations.
  – Salesforce Object Search Language (SOSL) search operation.
  – Utility (GetUserInfo, GetServerTimestamp, and SendEmail) 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. See Refresh Integration Metadata.
What Application Version Is Supported?

For information about which application version is supported by this adapter, see the Oracle Integration Adapters Certification Matrix under section Oracle Integration Adapters Certification at the top of the page.

Oracle Integration Adapters Certification Matrix

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.

Workflow to Create and Add a Salesforce Adapter Connection to 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.

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

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
<th>More Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>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.</td>
<td>Create a Salesforce Adapter Connection</td>
</tr>
<tr>
<td>2</td>
<td>Create the integration. When you do this, you add trigger and invoke connections to the integration.</td>
<td>Create Integrations and Add the Salesforce Adapter Connection to an Integration</td>
</tr>
<tr>
<td>3</td>
<td>Map data between the trigger connection data structure and the invoke connection data structure.</td>
<td>Map Data of Using Integrations in Oracle Integration</td>
</tr>
<tr>
<td>Step</td>
<td>Description</td>
<td>More Information</td>
</tr>
<tr>
<td>------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>4</td>
<td>(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).</td>
<td>Manage Lookups of <em>Using Integrations in Oracle Integration</em></td>
</tr>
<tr>
<td>5</td>
<td>Activate the integration.</td>
<td>Manage Integrations of <em>Using Integrations in Oracle Integration</em></td>
</tr>
<tr>
<td>6</td>
<td>Monitor the integration on the dashboard.</td>
<td>Monitor Integrations of <em>Using Integrations in Oracle Integration</em></td>
</tr>
<tr>
<td>7</td>
<td>Track payload fields in messages during runtime.</td>
<td>Assign Business Identifiers for Tracking Fields in Messages and Manage Business Identifiers for Tracking Fields in Messages of <em>Using Integrations in Oracle Integration</em></td>
</tr>
<tr>
<td>8</td>
<td>Manage errors at the integration level, connection level, or specific integration instance level.</td>
<td>Manage Errors of <em>Using Integrations in Oracle Integration</em></td>
</tr>
</tbody>
</table>
Create a Salesforce Adapter Connection

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

Topics:

• Prerequisites for Creating a Connection
• Create a Connection
• Upload an SSL Certificate
• Refresh Integration Metadata

Prerequisites for Creating a Connection

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

• Understand Salesforce constraints. See Understand Salesforce 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. Visit https://www.salesforce.com.
• Create the type of WSDL to use in an integration:
  – Enterprise WSDL. See Create the Salesforce Adapter Enterprise WSDL.
  – Outbound messaging WSDL. See Create the Salesforce Adapter Outbound Messaging WSDL.
  – Custom WSDL: See Create the Salesforce Adapter Custom WSDL.
• If you want to use the Salesforce OAuth Three-Legged or Resource Owner Password Credentials security policy, create a connected application in Salesforce.com. See Use the Salesforce OAuth Three-Legged or Resource Owner Password Credentials Security Policy.

Understand Salesforce 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.
• Not all the push topic queries are supported by Salesforce.
• Client applications must adhere to Salesforce’s SOAP API support policy and backward compatibility terms.
Create 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 Configure Connection Properties.

Create 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](http://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 Trigger Outbound Messaging Page.

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.
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](http://www.salesforce.com) and replace the dummy URL you entered in Step 3 with the real endpoint URL.

Create 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 Invoke Basic Information Page.

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

Use the Salesforce OAuth Three-Legged or Resource Owner Password Credentials Security Policy

To use the Salesforce OAuth Three-Legged or Resource Owner Password Credentials security policy, you must create a connected application in Salesforce.com.

1. Log in to your Enterprise, Unlimited, or Developer Edition Salesforce.com account.
2. In the upper-right corner, click your username dropdown, and select **Setup**.
3. Navigate to **App Setup > Create > Apps**.
4. In the **Connected Apps** section, click **New**. The New Connected App page appears.

5. In the **Basic Information** section, enter your basic information details, and select the **Enable OAuth Settings** checkbox in the API section.

6. In the **Callback URL** field, enter the fully qualified domain name of your server using the HTTPS protocol, and add the following text to the URL: `agent/oauth/callback`.

   For example,

   https://www.your_server.com/agent/oauth/callback

7. From the **Available OAuth Scopes** list, add the following scopes to **Selected OAuth Scopes**:
   - Access and manage your data (api)
   - Access your basic information (id)
   - Perform requests on your behalf at any time (refresh_token)

   **Note:**
   These are the minimum scopes required to configure the Salesforce OAuth Three-Legged security policy for your Salesforce Adapter connection. For the Resource Owner Password Credentials security policy, **api** and **id** are the minimum scopes required.

8. Click **Save**, and then click **Continue**. Once the application is saved, Salesforce populates the API section with the following IDs:
   - Consumer key (Client ID)
   - Consumer secret (Client secret)

   **Note:**
   - Use the client ID and client secret later when configuring security for your Salesforce Adapter connection. See **Configure Connection Security**.
   - Every connected application permits five distinctive approvals per user. Once a sixth approval is created, the oldest approval is revoked. See **Error When a Connected Application Attempts to Access Your Salesforce Data**.

---

**Create 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 navigation pane, click **Integrations**, then click **Connections**.
2. Click **Create**.

**Note:**
You can also create a connection in the integration canvas of:
- An orchestrated integration (See Define Inbound Triggers and Outbound Invokes.)
- A basic routing integration (See Add a Trigger (Source) Connection.)

The Create Connection — Select Adapter dialog is displayed.

3. 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 Create New Connection dialog is displayed.

4. 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, **Sales Opportunity**).
   - 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 Service Cloud (RightNow) Adapter as only an invoke, but drag the adapter to the trigger section).
   - Enter an optional description of the connection.
5. Click Create.

Your connection is created and you are now ready to configure connection details, such as email contact, connection properties, security policies, connection login credentials, and (for certain connections) agent group.

Add 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.

Configure 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 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. See Create the Salesforce Adapter Enterprise WSDL.
3. Click OK.
Configure Connection Security

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

1. Click **Configure Security**.

2. From the **Security Policy** list, select the security policy.
   - **Salesforce Username Password Policy**
   - **Authorization Code Credentials** (Salesforce OAuth 3-legged)
   - **Resource Owner Password Credentials**

3. If you select **Salesforce Username Password Policy**.
   a. In the **Username** field, enter the username.
   b. In the **Password** field, enter the password (combination of the password and security token).

   The security token must be appended to the end of the password. Salesforce.com APIs generate the security token at the time of password reset. For example, if the security token generated by Salesforce.com is SSSSSSSSSSS and the password is password, you must enter passwordSSSSSSSSSSSS to log in. Obtain the security token by changing the password or resetting the security token through the Salesforce.com user interface. You receive the security token through the email address registered with Salesforce.com. This token is valid until you reset the security token for the respective account or change the password.
   c. In the **Confirm Password** field, re-enter the password a second time for confirmation.

4. If you select **Authorization Code Credentials**.
   a. In the **Client Id** field, enter the consumer key/client ID that you obtained when creating a connected application. See Use the Salesforce OAuth Three-Legged or Resource Owner Password Credentials Security Policy.
   b. In the **Client Secret** field, enter the client secret key that you obtained when creating a connection application. See Use the Salesforce OAuth Three-Legged or Resource Owner Password Credentials Security Policy.
   c. In the **Confirm Client Secret** field, re-enter the client secret a second time for confirmation.
   d. In the **Scope** field, enter id api refresh_token. These are the minimum scopes required to configure a connection. Ensure that your connected application has these scopes configured. See Use the Salesforce OAuth Three-Legged or Resource Owner Password Credentials Security Policy.

   **Note:**

   The **Scope** field is not marked with an asterisk (*). This incorrectly implies that this field is not mandatory. This field is mandatory.

   e. Click **Provide Consent**. You are redirected to the Salesforce login page.
f. Log in to the Salesforce account with your credentials. Once you successfully log in, the following message appears:

Access Allowed! Please switch back to connection page to proceed with testing and saving connection configuration.

5. If you select Resource Owner Password Credentials.
   a. In the Client Id field, enter the consumer key/client ID that you obtained when creating a connected application. See Use the Salesforce OAuth Three-Legged or Resource Owner Password Credentials Security Policy.
   b. In the Client Secret field, enter the client secret key that you obtained when creating a connection application. See Use the Salesforce OAuth Three-Legged or Resource Owner Password Credentials Security Policy.
   c. In the Confirm Client Secret field, re-enter the client secret a second time for confirmation.
   d. In the Username field, enter the username.
   e. In the Password field, enter the password.
   f. In the Confirm Password field, re-enter the password a second time for confirmation.

Note:
When switching the connection from one Salesforce environment to another or switching the security policy, you must reactivate the integration to point to the correct environment.

Test the Connection

Test your connection to ensure that it is successfully configured.

1. In the upper right corner of the page, click Test.
2. If your adapter connection uses a WSDL, you are prompted to select the type of connection testing to perform:
   • **Validate and Test**: Performs a full validation of the WSDL, including processing of the imported schemas and WSDLs. Complete validation can take several minutes depending on the number of imported schemas and WSDLs. No requests are sent to the operations exposed in the WSDL.
   • **Test**: Connects to the WSDL URL and performs a syntax check on the WSDL. No requests are sent to the operations exposed in the WSDL.

   If successful, the following message is displayed and the progress indicator shows 100%.
   Connection connection_name was tested successfully.

3. If your connection was unsuccessful, an error message is displayed with details. Verify that the configuration details you entered are correct.
4. When complete, click Save, then click Close.
Upload 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, an exception is thrown. In that case, you must upload the appropriate certificate. A certificate enables Oracle Integration to connect with external services. If the external endpoint requires a specific certificate, request the certificate and then upload it into Oracle Integration.

To upload an SSL certificate:

1. In the navigation pane, click **Integrations**, then click the < arrow next to **Designer**.
2. Click **Settings** > **Certificates**.

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. 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.

3. Click **Upload** at the top of the page.
4. In the Upload Certificate dialog box, select the certificate type. Each certificate type enables Oracle Integration to connect with external services.

- **Trust Certificate**: Use this option to upload a trust certificate.
  a. Enter a unique alias for the certificate.
  b. Click **Browse**, then select the trust file (for example, `.cer` or `.crt`) to upload.

- **Message Protection Certificate**: Use this option to upload a keystore certificate with SAML token support. Create, read, update, and delete (CRUD) operations are supported on this type of certificate.
  a. Enter a unique alias for the certificate.
  b. Click **Browse**, then select the certificate file (`.cer` or `.crt`) to upload.

- **Identity Certificate**: Use this option to upload a certificate for two-way SSL communication.
  a. Click **Browse**, then select the keystore file (`.jks`) to upload.
  b. Enter the password of the keystore being imported.
  c. Enter the comma-separated list of aliases from the keystore being imported.
  d. Enter the comma-separated list of passwords corresponding to key aliases.
e. If you want to display the passwords in clear text, select **Show Key Password(s)**. This enables you to ensure that you are correctly entering a list of keystore passwords.

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.

---

**Refresh 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 navigation pane, click **Integrations**, then click **Connections**.
2. Locate the connection to refresh.
3. 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.
Add the Salesforce Adapter Connection to an Integration

When you drag the Salesforce Adapter into the trigger or invoke area of an integration, the Adapter Endpoint Configuration Wizard appears. This wizard guides you through the configuration of the Salesforce Adapter endpoint properties.

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

Topics:

• Basic Info Page
• Trigger Outbound Messaging Page
• Trigger Response Page
• Trigger Callback Response Page
• Invoke Basic Information Page
• Invoke Action Page
• Invoke Operations Page
• Invoke Headers Page
• Invoke Apex Operations Page
• Summary Page
• Perform Salesforce Adapter Postconfiguration Tasks

Basic Info Page

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

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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,="#;834 or right(now4)  
  • Multibyte characters |
Trigger Outbound Messaging Page

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. See Create the Salesforce Adapter Outbound Messaging WSDL.

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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. 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](http://www.salesforce.com) and replace the dummy URL you previously entered with the real endpoint URL. |

Trigger Response Page

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

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send a response</td>
<td>Deselect if no callback response is required.</td>
</tr>
<tr>
<td>Configuration a Successful Callback Response</td>
<td>Select <strong>Configure</strong> to configure the operation and business objects to use for a successful callback response.</td>
</tr>
<tr>
<td>Configuration a Failure Callback Response</td>
<td>Select <strong>Configure</strong> 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.</td>
</tr>
</tbody>
</table>
Trigger Callback Response Page

Enter the Salesforce trigger callback response values for your integration.

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

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Edit</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Header</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Reset</strong></td>
<td>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.</td>
</tr>
</tbody>
</table>

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, Process, 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.
- **Utility**: Represents the Send Email operation of Salesforce.com to perform in the Salesforce Cloud application. It composes a single email message or mass email messages (Send Mass Email) and sends the email(s) from the Salesforce.com organization.
**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.

**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.

---

### Invoke Basic Information Page

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

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
</table>
| **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, #;838 or rightnow4)  
  - 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. |
Invoke Action Page

Select the type of action to perform in the Salesforce.com application:

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
</table>
| What would you like to do in Salesforce.com? | • Query Information: Fetches data from Salesforce.com objects such as accounts, opportunities, and so on using the Retrieve, Query, and Search operations.  
• Create, Update Or Delete Information: Performs operations for account creation, opportunity status update, and so on. This is the default operation.  
• Perform Bulk Data Operations: Inserts, updates, upserts, or deletes a large volume of records by using the Salesforce.com Bulk API.  
Note: This option is enabled in the App Driven Orchestration integration style and is disabled in the Basic Routing integration style. See Understand Integration Patterns.  
• Perform Core or Unity Operations: Invokes Salesforce.com operations such as Merge, Send Email, and so on.  
• Invoke Apex Web Services: Consumes custom methods using Apex and exposed as SOAP web services on force.com. |

Invoke Operations Page

Enter the Salesforce invoke operation values for your integration.

Select the operation type to perform in the Salesforce.com application:

• Query Information
• Create, Update Or Delete Information Operation Type
• Perform Bulk Data Operations
• Perform Core or Utility Operations

Query Information

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
</table>
| If you selected **Query Information** on the Action page 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 options are displayed. | • **Query**: Executes a query against specific criteria and returns data matching that criteria. See Retrieve the Name and IDs from Account Records with the queryMore Operation.  
  Deleted and achieved records:  
  – **Exclude**: If you selected the **Query** operation and **Exclude** option, a query is executed that returns records that are not deleted from your Salesforce application account.  
  – **Include**: If you selected the **Query** operation and **Include** option, a query is executed that returns the same data as the query operation, along with deleted records present in the recycle bin.  
• **Retrieve**: Retrieves accounts from Salesforce application.  
• **Search**: Returns records from the Salesforce application. You can specify binding parameters to dynamically provide a search string as input to your search operation. |

**Enter a Salesforce Object Query Language (SOQL) 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"
```

**Click here for SOQL Query Reference**
Contains a list of SOQL query samples.
<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binding Parameters</td>
<td>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 = &quot;orgId&quot;. This query displays a binding parameters text box in which to enter a test value for orgId.</td>
</tr>
<tr>
<td>Test My Query</td>
<td>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.</td>
</tr>
</tbody>
</table>

### Create, Update Or Delete Information Operation Type

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>If you selected <strong>Create, Update Or Delete Information</strong> on the Action page, the following option is displayed.</td>
<td><strong>Create, Update, Upsert, or Delete Information</strong>: Represents the create, read, update, upsert, delete, or destroy operation to perform on Salesforce business objects. Each letter maps to a standard SQL statement, HTTP method, or DDS operation. <strong>Note</strong>: For the <strong>Update</strong> operation, an external ID cannot be mapped to an ID.</td>
</tr>
</tbody>
</table>
Perform Bulk Data Operations

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create</td>
<td></td>
</tr>
<tr>
<td>Delete</td>
<td></td>
</tr>
<tr>
<td>FinalBatch</td>
<td></td>
</tr>
<tr>
<td>Get Batch Results</td>
<td>Requests the response data for the batches of its respective job. It takes the jobID and BatchId as inputs and downloads a file. The file contains the respective batch response data to Oracle Integration, the response file path, and the file name. This information is available to you in the response mappings.</td>
</tr>
<tr>
<td>Get Status For All Batches</td>
<td>Requests the details of the bulk job primarily for checking job status. Once the job completes, you can fetch the batch results using the Get Batch Results operation. It takes the jobID as input and provides details/status of all batches for the job ID provided.</td>
</tr>
<tr>
<td>Update</td>
<td></td>
</tr>
<tr>
<td>Upsert</td>
<td></td>
</tr>
</tbody>
</table>

A use case is provided for this operation. See Process Large Data Sets Asynchronously with Different Bulk Operations.
Perform Core or Utility Operations

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>If you selected <strong>Perform Core or Utility Operations</strong> on the Action page, the following options are displayed.</td>
<td>• <strong>Convert Lead</strong>: Converts a lead into an account, a contact, or an opportunity in Salesforce.com.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Get Deleted</strong>: Retrieves deleted IDs of the Salesforce record of a selected object for a limited time period (maximum of one month).</td>
</tr>
<tr>
<td></td>
<td>• <strong>Get Server Timestamp</strong>: Retrieves the server timestamp (Coordinated Universal Time (UTC) time zone). This operation does not take any input data. This operation returns an element named timestamp of type <strong>dateTime</strong> that contains the server timestamp value.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Get Updated</strong>: Retrieves IDs of an updated Salesforce record of a selected object for a limited time period (maximum of one month).</td>
</tr>
<tr>
<td></td>
<td>• <strong>Get User Info</strong>: Retrieves information about the current user being used in the Salesforce connection configuration. This operation does not take any input data. This operation returns user information such as <strong>userId</strong>, <strong>userFullName</strong>, <strong>userEmail</strong>, <strong>profileId</strong>, <strong>roleId</strong>, <strong>organizationId</strong>, and so on.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Merge</strong>: This operation merges up to three records into one record in Salesforce.com.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Process</strong>: Related to the approval process in Salesforce.com. An approval process automates how records are approved, rejected, or removed in Salesforce.com. With the help of the Salesforce Adapter, you can use two process modes to submit the approval request and process the requests approval action.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Process mode:</strong></td>
</tr>
<tr>
<td></td>
<td>– <strong>ProcessSubmitRequest</strong></td>
</tr>
<tr>
<td></td>
<td>– <strong>ProcessWorkitemRequest</strong></td>
</tr>
<tr>
<td></td>
<td>• <strong>Undelete</strong>: Restores Salesforce.com records from the recycle bin.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Send Email</strong>: Triggers a single email message request to Salesforce. This operation takes most standard email attributes, <strong>templateIds</strong>, <strong>targetObjectIds</strong>, and other attributes as input. This operation responds with the Boolean element <strong>Success</strong> that defines the status of the message triggered. If the email message is successfully triggered, it responds with <strong>true</strong>. Otherwise, it responds with <strong>false</strong>.</td>
</tr>
<tr>
<td>Element</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>Send Mass Email</strong></td>
<td>Triggers a mass email message request to Salesforce. This operation takes most standard email attributes, <code>templateIds</code>, multiple <code>targetObjectIds</code>, and other attributes as input. This operation returns the Boolean element <strong>Success</strong> that defines the status of the message triggered. If the email message is successfully triggered, it responds with <code>true</code>. Otherwise, it responds with <code>false</code>.</td>
</tr>
</tbody>
</table>

**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" (two underscores).
- **Standard**: Displays business objects delivered as part of the Salesforce application.
- **Recommended**: Displays the most frequently used objects.

**Select Business Objects**
Select a single or multiple business objects to include in the operation. If the operation supports multiple objects, you can select up to ten objects for one operation.

**Use Default Header**
If this checkbox is selected, the Headers page is skipped. Default headers based on the operation you selected on the Operations page are applicable. You can deselect this checkbox if you want to use a header other than the default.

---

**Invoke Headers Page**

Enter the Salesforce invoke header values for your integration.

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](http://www.salesforce.com) and specify the specific name of the header property in the search utility.
### Note:
The Headers page is visible only if the Use Default Header checkbox is deselected on the Operations page.

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>All Or None Header</strong> (request header)</td>
<td>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.</td>
</tr>
<tr>
<td>- Integration flows created before version 16.4.1:</td>
<td>- If set to true and the response contains error elements, it maps to UnexpectedErrorFault. This is also the case if All Or None Header is set to false (that is, unselected).</td>
</tr>
</tbody>
</table>
| - 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.

<table>
<thead>
<tr>
<th>Allow Field Truncation Header (request header)</th>
<th>Specifies the truncation behavior for the following fields (each are string data types):</th>
</tr>
</thead>
</table>
| - anyType                                     | - email  
- picklist  
- encryptedstring  
- textarea  
- multipicklist  
- phone  
- string  |

Set Allow Field Truncation 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.
<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Assignment Rule Header</strong> (request header)</td>
<td>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 Use Default Rule value.</td>
</tr>
<tr>
<td>• <strong>Assignment Rule Id</strong>: 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.</td>
<td></td>
</tr>
<tr>
<td>• <strong>Use Default Rule</strong>: If set to true, the default (active) assignment rule is used. If set to false, the default (active) assignment rule is not used.</td>
<td></td>
</tr>
<tr>
<td><strong>Duplicate Rule Header</strong></td>
<td>Specifies the duplicate management behavior of the Salesforce application operations. The behavior of this parameter is handled in a mutually exclusive way. This header has three properties:</td>
</tr>
<tr>
<td>• <strong>Allow Save</strong>: This property is selected by default when configuring a Salesforce endpoint.</td>
<td></td>
</tr>
<tr>
<td>– <strong>True</strong> (if selected): Creates/updates duplicate records. The selection of this property disables the selection of Include Record Details and Run As Current User.</td>
<td></td>
</tr>
<tr>
<td>– <strong>False</strong> (if deselected): Prevents created and updated duplicate records from being saved.</td>
<td></td>
</tr>
<tr>
<td>• <strong>Include Record Details</strong>: This option is selectable when Allow Save is deselected. This option enables you to get record details of existing duplicates.</td>
<td></td>
</tr>
<tr>
<td>– <strong>True</strong> (if selected): Retrieves duplicate error details and duplicate record details that impact the transaction.</td>
<td></td>
</tr>
<tr>
<td>– <strong>False</strong> (if deselected): Retrieves duplicate error details that do not include duplicate records.</td>
<td></td>
</tr>
<tr>
<td>• <strong>Run As Current User</strong>: Enforces the sharing of rules for the current user.</td>
<td></td>
</tr>
<tr>
<td>– <strong>True</strong> (if selected): Runs duplicate rules for the current user and ignores the rules not available to them.</td>
<td></td>
</tr>
<tr>
<td>– <strong>False</strong> (if deselected): Applies shared rules for the current user.</td>
<td></td>
</tr>
</tbody>
</table>

A use case that describes how to use the Duplicate Rule Header property is provided. See Allow and Prevent Duplicate Headers with the Duplicate Rule Header Property.
### Email Header (request header)

Specifies whether or not to send a notification email. You can set the following properties:

- **Trigger Auto Response Email**
  - **true**: Triggers automatic response rules for leads and cases.
  - **false**: Automatic response rules for leads and cases are not triggered.

- **Trigger Other Email**
  - **true**: The email is triggered outside the organization.
  - **false**: The email is not triggered outside the organization.

- **Trigger User Email**
  - **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.

### Mru Header (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 **Mru Header** to update the list. Using this header can negatively impact performance. Set **Update Mru** 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.

### Package Version Header (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`).

### Query Options (request header)

Specifies the batch size for queries. The minimum and default value is 200 and the maximum value is 2000.

---

**Invoke Apex Operations Page**

Specify the following values on the Salesforce invoke Apex operations page.

If you selected **Invoke Apex Web Services** on the Action page, the following options are displayed.
### Summary Page

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

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Summary</td>
<td>Displays a summary of the configuration values you defined on previous pages of the wizard.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>To return to a previous page to update any values, click the appropriate tab in the left panel or click Back. Click Cancel to cancel your configuration details.</td>
</tr>
</tbody>
</table>

### Perform Salesforce Adapter Postconfiguration Tasks

After activating your integration, you must update the outbound message for the Salesforce Adapter to send messages to Oracle Integration. 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.

   ![Note]

   Remove ?wsdl from the endpoint URL. See Errors When the Salesforce Endpoint is Configured as a Trigger.

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.
Implement Common Patterns Using the Salesforce Adapter

You can use the Salesforce Adapter to implement the following common patterns.

**Topics:**
- Process Large Data Sets Asynchronously with Different Bulk Operations
- Use Bulk Response Operations in an Integration
- Retrieve the Name and IDs from Account Records with the queryMore Operation
- Send Multiple Records in a Single Message
- Allow and Prevent Duplicate Headers with the Duplicate Rule Header Property

**Process Large Data Sets Asynchronously with Different Bulk Operations**

The Salesforce Bulk API enables you to handle huge data sets asynchronously with different bulk operations. For every bulk operation, the Salesforce application creates a job that is processed in batches.

A job contains one or more batches in which each batch is processed independently. The batch is a nonempty CSV/XML/JSON file that is limited to 10,000 records and is less than 8 MB in size. Because the batches are processed in parallel, no execution order is followed. A batch can contain a maximum of 10,000,000 characters in which 5,000 fields in a batch are allowed with a maximum of 400,000 characters for all its fields and 32,000 characters for each field.

This use case discusses about how to configure the Salesforce Adapter to create a large number of account records in Salesforce Cloud.

To perform this operation, you create FTP Adapter and Salesforce Adapter connections in Oracle Integration.

In this use case, a CSV file is used as input, but you can also use other format files. The Salesforce Adapter transforms the file contents into a Salesforce-recognizable format.
1. Create a scheduled orchestration integration.

2. Drag an FTP Adapter into the integration canvas.

3. Configure the FTP Adapter as follows:
   a. On the Operations page, select **Download File** from the **Select Operation** list.
   b. Select **ASCII** from the **Select a Transfer Mode** list.
   c. Provide the input directory, file name, and download directory.
   d. Review your selections on the Summary page.

4. Drag a stage file action into the integration canvas below the FTP Adapter. The stage file action helps the Salesforce Adapter fetch data in the form of segments (a single segment contains 200 records).
   a. On the Basic Info page, provide a name.
   b. On the Configure Operations page, select **Read Files in Segments** from the **Choose Stage File Operation** field.
   c. Specify the XPath for the file name in the **Specify the File Name** field.
   d. Specify the directory name in the **Specify the Directory to read from** field (downloaded to Oracle Integration using FTP).
   e. On the Scheme Options page, select **Create a new schema from a CSV file** in the **Do you want to create a new schema or select an existing one** list.
   f. On the Format Definition page, click **Choose File** and upload the sample CSV file in the **Select a New Delimited Data File** field.
   g. Review your selections on the Summary page.

5. Drag a Salesforce Adapter inside the stage file action and configure it with the required **BULK** operation.
   a. On the Basic Info page, provide a name.
   b. Select **standard applications delivered by Salesforce.com**.
   c. On the Action page, select **Perform Bulk Data Operations**.
d. On the Operations page, select the required operation (for this example, Create) in the Select an Operation field.

e. Select the required business objects (for this example, Account) in the Select Business Objects field.

f. Review your selections on the Summary page.

6. In the mapper, map the stage file action elements to the Salesforce input payload.

7. Drag a Salesforce Adapter below the stage file action and configure it with the Final Batch operation.

   a. On the Basic Info page, provide a name.

   b. On the Action page, select Perform Bulk Data Operations.

   c. On the Operations page, select Final Batch in the Select an Operation field.

   **Note:** You cannot select the object on which to perform the operation because this Salesforce Adapter connection performs the activity for the operation that was configured for the Salesforce endpoint inside the stage file action.

8. Drag an FTP Adapter connection below the Salesforce Adapter and configure the FTP endpoint to write the Salesforce final batch response to a file for future use.

   a. On the Basic Info page, provide a name.

   b. On the Operations page, select Write File from the Select Operation list.

   d. Review your selections on the Summary page.
c. Select ASCII from the Select a Transfer Mode list.

d. Specify the output directory, file name pattern, and download directory.

e. On the Scheme page, select XML schema (XSD) document (to describe the XML message) from the Which one of the following choices would be used to describe the structure of the file contents list.

f. On the File Contents-Definition page, click Choose File and upload the schema source file in the Select a New File field.

g. Review your selections on the Summary page.

9. In the mapper, map the Salesforce final batch response to the FTP write request. The completed integration looks as follows.

10. Specify the tracking variable.

   a. Click Tracking.

   b. Drag the required variable to use for the tracking reference (for this example, startTime is selected).
Activate the integration.

After successful activation, you can submit the integration and monitor the runtime in Oracle Integration. Once you receive the reference bulk job ID in the final batch response, you can also get the job status using the Get Status for all Batches operation available under the Bulk Operation type.

Salesforce provides documentation on bulk operations. See https://resources.docs.salesforce.com/sfdc/pdf/api_asynch.pdf.

Use Bulk Response Operations in an Integration

This use case provides an overview of how to retrieve the response for a job created by a bulk operation (create, update, upsert, and delete).

This use case uses the following bulk operations.

- Get Status For All Batches: Requests the details of the bulk job primarily for checking job status. Once the job completes, you can fetch the batch results using the Get Batch Results operation. It takes the jobId as input and provides details/status of all batches for the job ID provided.

- Get Batch Results: Requests the response data for the batches of its respective job. It takes the jobId and BatchId as inputs and downloads a file. The file contains the respective batch response data to Oracle Integration, the response file path, and the file name. This information is available to you in the response mappings.

The functional overview of the integration is as follows:
Create an Orchestrated Integration and Perform the Initial Connection Configuration and Mapping

- **Note:**
  This use case assumes you have already configured a SOAP Adapter, Oracle Integration, and FTP Adapter on the Connections page.

1. On the Integrations page, click **Create**.
2. Select the **Orchestration** pattern, click **Select**, then select **Application Event or business object** as the option to trigger this integration.
3. Drag the SOAP Adapter to the trigger section of the integration canvas.
4. Complete the Adapter Endpoint Configuration Wizard pages for the SOAP Adapter by selecting appropriate operation and header values.
5. Drag the Oracle Integration below the SOAP Adapter in the integration canvas.

6. Complete the Adapter Endpoint Configuration Wizard pages for the Oracle Integration by specifying the following:
   - Name the connection (for this example, named SFDC_GetAllBatchStatus).
   - Select the **Perform Bulk Data Operations** operation type.
   - Select the **Get Status For All Batches** operation.

7. Open the mapper and map the source SOAP request input element to the target Salesforce **getAllBatchStatus** input payload. For example, map the `input1` element to the `jobId` element.

![Map to SFDC_GetAllBatchStatus](image)

**Create a For Each Action in the Integration**

1. Drag a **For Each** action immediately below the **SFDC_GetAllBatchStatus** Oracle Integration.

2. Name the action, then drag `batchInfo` (the response of `getallbatches`) to the **Repeating Element** field.

3. In the **Current Element Name** field, enter a name (for this example, `TempBatch`), then click **Done**.

![For Each Action](image)

4. Drag a second Oracle Integration connection to the workspace inside the **For Each** action.
5. Complete the Adapter Endpoint Configuration Wizard pages for the Oracle Integration by specifying the following:
   • Naming the connection (for this example, SFDC_GetBatchResults).
   • Specifying the Perform Bulk Data Operations operation type.
   • Specifying the Get Batch Results operation.

6. In the response mapper for getBatchResults, map the source id element under TempBatch to the target batchId and the source jobId element under TempBatch to the target jobId.
7. Validate the mappings, then close the mapper and save your changes when prompted.

Add a Stage File Action to the For Each Action

1. Drag a Stage File action inside the For Each action and below the SFDC_GetBatchResults Oracle Integration. The stage file action can read (and remove any trailer), write, zip, unzip, and list files in a staged location known to Oracle Integration.

   ![Diagram showing the placement of Stage File action inside For Each action]

   The Configure Stage File Action wizard is displayed.

2. Provide a name, then click Next.

3. On the Configure Operation page, select Read Entire File from the Choose Stage File Option list, then click the Select Expression icon to specify the file name.

4. Drag the source fileName element to the Expression field, then click the Expression Summary Refresh icon to display the name. This action provides a file name to receive in the SFDC_GetBatchResults response.
5. Click **Save**, then **Exit Expression Builder**.

6. Click the **Select Expression** icon to specify the directory from which to read the file.

7. Drag the source **dirPath** element to the **Expression** field, then click the **Expression Summary Refresh** icon to display the name.

8. Click **Save**, then **Exit Expression Builder**.

9. Review your selections, and click **Next**.
10. On the Schema Options page, select **Create a new schema from a CSV file**, and click **Next**.

11. On the Format Definition page, browse for and select a CSV file that maintains the response file structure (for this example, *result.csv*), and click **Next**.

![Format Definition page](image1)

12. On the Summary page, review your selections, and click **Done**.

**Drag an FTP Adapter Connection Inside the For Each Action**

1. Drag an FTP Adapter connection into the **For Each** action and below the **Stage File** action you just configured to read the batch results file.

2. In the Adapter Endpoint Configuration Wizard, provide an endpoint name, and click **Next**.

3. On the Operations page, specify the following details:
   - Select **Write File** from the **Select Operation** list.
   - Specify an output directory for the file.
   - Specify a file name pattern.

![Operations page](image2)

4. On the Schema page, create a schema using a sample bulk file (for this example, *response.csv*).
5. Select a CSV file that maintains the response file structure, and click Next. For this example, the same CSV file used for the stage file action is used.

6. Review your selections on the Summary page, and click Done. The integration looks as follows:
Configure Mapping for the FTP Adapter Connection

1. Click the mapper above the FTP Adapter.
2. Map the stage response elements to the FTP request elements.

3. Click the mapper below the FTP Adapter.
4. Map the SOAP Adapter response with the **getAllBatchStatus** response (the **SFDC_GetAllBatchStatus** response elements to the SOAP response object).

5. Validate the mapping, and click **Close**.

**Add a Tracking Variable**

1. From the **menu in the upper-right corner, select Tracking.**
2. Configure a tracking variable, and click **Done.**

The completed integration looks as follows without any errors and warnings.
3. Save and close the integration.
   The Integrations page is displayed.

Activate the Integration

1. At the end of the row for the integration, click the **Switch to Activate** icon.
2. If needed, check the **Enable tracing** and **Include payload** check boxes.
3. Click **Activate**.
   The integration is deployed and the web service is ready to accept requests.
Retrieve the Name and IDs from Account Records with the queryMore Operation

This use case describes how to use the queryMore operation to retrieve the name and IDs for all account records from Salesforce.com. This use case uses the SOAP Adapter as the trigger to execute the integration.

The queryMore operation retrieves the next set of object records from the query operation.

The query function in SOQL retrieves a set of records and creates a server-side cursor that is represented in the queryLocator element. The queryMore operation processes the subsequent records in chunks and returns a newly-generated queryLocator. You typically call the queryMore operation repeatedly until all records in the result set have been processed.

The following important variables are used:

- **batchSize**: A request header that represents the number of records returned for each Salesforce trigger. The default value is 200, the minimum allowed size is 200, and the maximum is 2000.

- **queryLocator**: A value which points to the last record retrieved. Initially, it has no value. Therefore, you can initialize the local queryLocator with empty single quotes ("').

- **size**: A variable that contains the number of records that have been retrieved in the previous iteration.

- **done**: A boolean variable that indicates whether the source has more than the records you fetched. Its value is true if there are no records to fetch. Otherwise, it is false.

### Note:

- This orchestrated integration shows how to retrieve more than 200 records (when batchSize is set to 200) from Salesforce. You use the data by performing the required operation inside the while action loop. For validating, you can count the number of records returned for every Salesforce trigger.

- A Salesforce Aggregate query does not support the queryMore operation.

- While working with nested queries, if you use a queryMore call on a parent object, it invalidates all the child cursors in the previous result.

1. Create SOAP Adapter and Salesforce Adapter connections.
2. Create an app-driven orchestrated integration.
3. Drag the SOAP Adapter connection into the integration as a trigger connection.
4. Drag an assign action to the workspace next to the SOAP Adapter (for this example, named Init).
This assign action declares and initializes the $done and $querylocator variables locally.

These variables are initialized with false and "" (no value within the single quotes) values for $done and $querylocator, respectively. The $countRecords variable counts the records retrieved and initialized to 0.

5. Drag a while action below the assign action and set the condition to $done= ‘false’.

The while action loop iterates if the value of $done remains false.

6. Configure a Salesforce Adapter endpoint.
   a. Drag a Salesforce Adapter as an invoke connection inside the while action.
   b. On the Basic Info page, enter an endpoint name (for this example, named SFDC_Demo).
   c. On the Action page, select Query Information.
   d. On the Operations page, select Query in the Select an Operation field, and Exclude in the Deleted and archived records field.
   e. Enter a SOQL query in the text box and click Test My Query to verify the input query.

   Select id, name, BillingCity from Account

   where Select id, name, BillingCity retrieves the ID, name, and billing city from the Salesforce Account object for all the records.
f. On the Headers page, set the required **batch Size** (which allows a minimum value of 200) to the maximum value of 2000.

g. On the Summary page, review the entered values.

**Note:**

The scope of the configured Salesforce Adapter endpoint is limited to the while action. Therefore, you cannot access the Salesforce input and output values outside the while action.

7. In the mapper, map the local source **$queryLocator** variable to the target **QueryLocator** element of the request mapper. For every iteration, the input for the Salesforce endpoint is the **queryLocator** value, which is initialized with """" (empty value) in the first configured assign action (named **Init**). This is further updated dynamically with the Salesforce endpoint response in the second assign action (named **Update**).

8. Drag the assign action next to the Salesforce Adapter (for this example, named **Update**).

- Update the local variables (**$done** and **$queryLocator**) with Salesforce output variables.
  - For the **$done** variable:
  
  - For the **$queryLocator** variable:
• For the $countRecords variable:

   The count() function is available under Functions, categorized in Node-set. This takes Node-set as input and returns the count of nodes available in the Node-set. The record Node-set is passed to the count(). After mapping, the function call looks as follows: `count($SFDC_E/nsmpr3:queryResponse/nsmpr3:QueryResults/nsmpr3:records)

9. In the mapper, map the response. Enter an “Operation Successfully Completed” string as the status. The count of records available in the $countRecords variable are passed to outputString1 and outputString2, respectively.

10. Update the tracker with any variable for tracking reference.

    The completed integration looks as follows.
Send Multiple Records in a Single Message

The Salesforce Adapter Send Outbound Message action can send multiple records in a single message at specific times, particularly when a bulk data import occurs within Salesforce. Up to 100 records can be sent in a single message request. For example, if an outbound message trigger is enabled for Account create and Account is created in bulk in Salesforce.com, there are multiple records (up to 100 records) coming from Salesforce in a single message.

In the following scenario, the integration receives a single message containing multiple records. To support this scenario, you must configure the mapper when the request record shows a single occurrence, but contains multiple records. For this example, the steps to map the Notification element when it shows a single occurrence are shown.

1. Click the Mapper icon in your integration.
2. Click the parent element in the Target section (for example, the element Account for Salesforce Account).
   The Build Mappings page is displayed.
3. Add a for-each loop to the parent element.
   a. Click the Mapping Components tab in the bottom left corner.
   b. Expand XSL Elements.
   c. Drag and drop the for-each function to the parent element of the Target section (for this example, Account).
4. Drag and drop the **Notification** element to the **select** section inside the for-each loop.

5. Validate and close the Build Mappings page.

6. Complete the mappings.

   When complete, the mapper looks as follows:
Allow and Prevent Duplicate Headers with the Duplicate Rule Header Property

You can allow and prevent duplicate headers in messages with the **Duplicate Rule Header** property. This section provides an overview of the functionality that enables you to allow or prevent duplicate headers.

**Duplicate Rule**

The duplicate rule is part of a duplicate management feature provided by Salesforce to manage duplicate data. This feature enables you to define new duplicate rules and manage them to avoid duplicate records and data. Salesforce enables you to manage duplicate records through the SOAP API with the help of a header called **Duplicate Rule Header**.

The **Duplicate Rule Header** consists of three properties:

- **Allow Save**: Allows duplicate records when set to **true** (that is, selected) and prevents duplicate records from being saved when set to **false** (that is, deselected).

- **Include Details**: Returns fields and values along with the IDs of the records detected as duplicates when set to **true** (that is, selected) and retrieves only IDs for the records detected as duplicates when set to **false** (that is, deselected).

- **Run As Current User**: Enables or disables sharing rules for the current user. When set to **true** (that is, selected), ensures that sharing rules for the current user are enforced when duplicate rules run. When set to **false** (that is, deselected), the sharing rules specified for the request are used.

**Duplicate Error**

The duplicate error carries information about the business error that occurs when a user creates a record that violates the duplicate rule.

- This is an extended data type of the error element.

- The duplicate result represents the details of duplicate records that are detected by the duplicate rule and information about those duplicate records.

**Duplicate Rule Header in Oracle Integration Cloud Salesforce Plugin**

The **Duplicate Rule Header** consists of the following components:

- **Duplicate rule**: The Salesforce plugin supports all header properties of the duplicate rule. These three properties are handled in a mutually exclusive way. That is, if the property **Allow Save** is set to **true** (that is, selected), the other two header properties **Include Details** and **Run As Current User** change to **false** (that is, deselected) and unavailable for editing. Similarly, if **Allow Save** is set to false (that is, deselected) the other two headers change to **true** (that is, selected) and are available for editing.

- **Duplicate result**: The **Duplicate Rule Header** results as an extended type of error (DuplicateError type).

  - This error type contains different fields that carry information related to duplicate records detected by the duplicate rule.
Because this is an extended error, the user must substitute the error with DuplicateError to make use of it.

The following implementation patterns show how to use Duplicate Rule Header:

- Allow Duplicate Records with the Duplicate Rule Header Property
- Prevent Duplicate Records with the Duplicate Rule Header Property

Allow Duplicate Records with the Duplicate Rule Header Property

You can allow duplicate records with the Duplicate Rule Header property.

Prerequisites:

- Configure and activate at least one matching rule under Duplicate Management in the respective Salesforce organization account.
- Configure and activate at least one duplicate rule under Duplicate Management for the respective active matching rule created. You can see the one created matching rule and the one duplicate rule in the Salesforce account that is configured to maintain the unique Phone field value for the Account object.

1. Create a new integration (for this example, a Basic Routing integration pattern is selected).

2. Configure a SOAP Adapter for the trigger connection.
   a. Drag and drop a SOAP Adapter connection to the trigger/source side.
   b. On the Basic Info page, specify an endpoint name, and click Next.
   c. On the Operations page, accept the default values, and click Next.
   d. On the Headers page, accept the default values, and click Next.
   e. On the Summary page, click Next.

3. Configure a Salesforce Adapter for the invoke connection.
   a. Drag and drop a Salesforce Adapter to the invoke/target side.
   b. On the Basic Info page, specify an endpoint name, and click Next.
   c. On the Action page, select Create, Update Or Delete Information.
   d. On the Operations page, select Create in the Select an Operation field.
   e. Select Account in the Select Business Objects table.
   f. On the Headers page, ensure that Allow Save is set to true under Duplicate Rule Header and All Or None Header is set to false.

   **Note:**

   When Allow Save is set to true (that is, selected), Include Record Details and Run As Current User remain set to false (that is, deselected), and vice versa.

   g. On the Summary page, review your selections, and click Done.

4. Configure request mapping.
   a. Click the Request mapping icon for editing.
b. Map the required elements.
   • **Account > Name** field: This is a mandatory field.
   • **Account > Phone** field: This is configured for a duplicate rule that does not allow duplicate phone numbers in your Salesforce account.

   ![Map the required elements](image)

   c. Click **Validate**, then click **Close**.

5. Configure response mapping.
   a. Click the **Response** mapper icon for editing.
   b. Map the required elements:
      • Map **Id** and **Success** to **output1** and **output2**, respectively.
      • Because **All Or None Header** is set to **false** (that is, deselected), you get a business error response in the **createResponse** tree instead of the **Fault** tree (unexpected error fault).

   This enables you to also map error elements. To map error elements, Salesforce returns different types of errors during runtime:
      • A generic **Error** element.
      • A **DuplicateError** element that is an extension of the generic **Error** element.
   c. Click **Repeat Element** to map both errors to the same element. The mappings are as follows:

   ![Map error elements](image)

   d. Click **Validate**, then click **Close**.

6. Add a business tracking variable.
   a. Click **Tracking**.
   b. Map the business identifier as a tracking variable.

7. When complete, save and activate the integration.

8. Once activated, use the endpoint URL to trigger the requests.
Prevent Duplicate Records with the Duplicate Rule Header Property

You can prevent duplicate records with the Duplicate Rule Header property.

Prerequisites:

- Configure and activate at least one matching rule under Duplicate Management in the respective Salesforce organization account.
- Configure and activate at least one duplicate rule under Duplicate Management for the respective active matching rule created. You can see the one created matching rule and the one duplicate rule in the Salesforce account that is configured to maintain the unique Phone field value for the Account object.

1. Create a new integration (for this example, a Basic Routing integration is selected).

2. Configure a SOAP Adapter for the trigger connection.
   a. Drag and drop a SOAP Adapter connection to the trigger/source side.
   b. On the Basic Info page, specify an endpoint name, and click Next.
   c. On the Operations page, accept the default values, and click Next.
   d. On the Headers page, accept the default values, and click Next.
   e. On the Summary page, click Next.

3. Configure a Salesforce Adapter for the invoke connection.
   a. Drag and drop a Salesforce Adapter to the invoke/target side.
   b. On the Basic Info page, specify an endpoint name, and click Next.
   c. On the Action page, select Create, Update Or Delete Information.
   d. On the Operations page, select Create in the Select an Operation field.
   e. Select Account in the Select Business Objects table.
   f. On the Headers page under Duplicate Rule Header, ensure that Include Record Details and Run As Current User are set to true (that is, selected) and Allow Save is set to false (that is, deselected). These settings configure Duplicate Rule Header to prevent duplicate records from being created.
   g. Set All Or None Header to false.
   h. On the Summary page, review your selections, and click Done.

4. Configure request mapping.
   a. Click the Request mapping icon for editing.
   b. Map the required elements.
      - Account > Name field: This is a mandatory field.
      - Account > Phone field: This is configured for a duplicate rule that does not allow duplicate phone numbers in your Salesforce account.
c. Click Validate, then click Close.

5. Configure response mapping.
   a. Click the Response mapper icon for editing.
   b. Map the required elements:
      • Map Id and Success to output1 and output2, respectively.
      • Because All Or None Header is set to false (that is, deselected), you get a business error response in the same path instead of an unexpected error fault.

This enables you to also map error elements. To map error elements, Salesforce returns different types of errors during runtime:
  • A generic Error element.
  • A DuplicateError element that is an extension of the generic Error element.

   c. Click Repeat Element to map both errors to the same element. The mappings are as follows:

   d. Click Validate, then click Close.

6. Add a business tracking variable.
   a. Click Tracking.
   b. Map the business identifier as a tracking variable.

7. When complete, save and activate the integration.

8. Once activated, use the endpoint URL to trigger the requests.
Troubleshoot the Salesforce Adapter

Review the following topics to learn about troubleshooting issues with Salesforce Adapter.

Topics:

• Error When a Connected Application Attempts to Access Your Salesforce Data
• Errors When the Salesforce Endpoint is Configured as a Trigger
• Invoke Oracle Integration When a Payload is Not Expected
• Id Field Identified with an Asterisk in the Request Mapper of an Upsert Operation is Not Mandatory
• Custom and Inbound WSDLs Unavailable after Migration

Additional integration troubleshooting information is provided. See Troubleshoot Oracle Integration in Using Integrations in Oracle Integration.

Error When a Connected Application Attempts to Access Your Salesforce Data

The following error can occur when a connected application to which you have granted access attempts to access your Salesforce data.

Could not refresh the access token. The access token response returned an unsuccessful status [400]

Every connected application permits five distinctive approvals per user. Once a sixth approval is created, the oldest approval is revoked. See Manage OAuth Access for Your Connected Apps.

Errors When the Salesforce Endpoint is Configured as a Trigger

Note the following errors that can occur when the Salesforce endpoint is configured as a trigger.

• The following error occurs when you enter the endpoint URL with ?wsdl in the outbound message for the Salesforce Adapter in the Salesforce application.

  org.xml.sax.SAXException: Bad envelope tag: definitions

  Solution: Remove ?wsdl from the endpoint URL.
The following error occurs when the endpoint URL updated in the Salesforce application mismatches with the configured Salesforce connection. For example, a Salesforce connection in Oracle Integration is configured with the sandbox account details. However, the endpoint URL (in the Salesforce application) is updated from the production account.

`org.xml.sax.SAXException: Bad envelope tag: HTML`

**Solution:** Update the endpoint URL that you configured for the Salesforce connection in Oracle Integration in the outbound message panel of the Salesforce application.

**Invoke Oracle Integration When a Payload is Not Expected**

If the Salesforce application invokes Oracle Integration with an Apex program and a payload is not expected, you must set the `Content-Length` header to 0 in the Apex program on the Salesforce side.

As per the HTTP/1.0 Draft RFC and also HTTP/1.1, which is backward compatible with HTTP/1.0, application clients must send a valid `Content-Length` header if the entity body must be sent. The `Content-Length` header can be zero or more.

In Oracle Integration, you either need to pass the `Content-Length` header (0 is a valid value) or pass `Transfer-Encoding`. A `Transfer-Encoding` value of `chunked` is assumed.

**Id Field Identified with an Asterisk in the Request Mapper of an Upsert Operation is Not Mandatory**

If you configure the Salesforce Adapter as an invoke (target) connection in an integration and select **Create, Update Or Delete Information** on the Action page and the operation type as **Upsert** in the Adapter Endpoint Configuration Wizard, when you get to the request mapper, the Id field is marked with an asterisk (*) for the **Upsert** operation. This incorrectly implies that this field is mandatory. This field is *not* mandatory.

**Custom and Inbound WSDLs Unavailable after Migration**

When an integration that includes a Salesforce endpoint as a trigger or invoke (configured with a custom application built using Apex classes and hosted on force.com) operation is exported and imported into another Oracle Integration environment (for example, from test to production) and the Salesforce connection already exists with the same name, the connection does not save the WSDL files as a part. Therefore, when you edit the Salesforce endpoint, you do not see the uploaded custom or outbound messaging WSDL files and you must upload these files again in the Adapter Endpoint Configuration Wizard.