

Oracle® Cloud

Using the Salesforce Adapter with Oracle Integration Generation 2



E85419-50
June 2023

The Oracle logo, consisting of the word "ORACLE" in white, uppercase letters, centered within a solid red square.

ORACLE®

Oracle Cloud Using the Salesforce Adapter with Oracle Integration Generation 2,

E85419-50

Copyright © 2017, 2023, Oracle and/or its affiliates.

Primary Author: Oracle Corporation

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, and MySQL are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface

Audience	vi
Documentation Accessibility	vi
Diversity and Inclusion	vi
Related Resources	vii
Conventions	vii

1 Understand the Salesforce Adapter

Salesforce Adapter Capabilities	1-1
Salesforce Adapter Restrictions	1-2
What Application Version Is Supported?	1-3
Salesforce Adapter Use Cases	1-3
Workflow to Create and Add a Salesforce Adapter Connection to an Integration	1-3

2 Create a Salesforce Adapter Connection

Prerequisites for Creating a Connection	2-1
Prerequisites for Oracle Integration Release 20.2.2.0.0 (200524.0200.35760) or Later	2-1
Identify the Instance Type of Your Salesforce Organization	2-2
Obtain the Custom Domain Name	2-2
Find Your Current Salesforce API Version	2-2
Prerequisite for Oracle Integration Releases Earlier Than 20.2.2.0.0	2-3
Generate the Salesforce Enterprise WSDL	2-3
Prerequisites to Use the Salesforce Username Password Security Policy	2-4
Prerequisites to Use the Salesforce OAuth Three-Legged (Authorization Code Credentials) or Resource Owner Password Credentials Security Policy	2-5
Prerequisites to Use Platform Events	2-6
Define Your Platform Event	2-6
Publish Platform Events	2-7
Prerequisite to Use Change Data Capture (CDC) Events	2-7
Select an Entity (Business Object)	2-7
Prerequisite to Use Outbound Messaging	2-8

Use Outbound Messaging with a Flow	2-8
Use Outbound Messaging with a Workflow	2-12
Prerequisite to Use Apex Web Services	2-14
Generate the Salesforce Adapter Custom WSDL	2-14
Create a Connection	2-14
Configure Connection Properties	2-15
Configure Connection Security	2-17
Test the Connection	2-19
Upload an SSL Certificate	2-19
Refresh Integration Metadata	2-21

3 Add the Salesforce Adapter Connection to an Integration

Basic Info Page	3-1
Trigger Action Page	3-2
Trigger Outbound Messaging Page	3-2
Trigger Streaming API Page	3-3
Trigger Response Page	3-5
Trigger Callback Response Page	3-5
Invoke Action Page	3-6
Invoke Operations Page	3-7
Invoke Headers Page	3-12
Invoke Apex Operations Page	3-15
Query and Template Parameters Page	3-17
Invoke Sample Request Payload Page	3-18
Invoke Sample Response Payload Page	3-19
Summary Page	3-20
Perform Salesforce Adapter Postconfiguration Tasks	3-20

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	4-16
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
Synchronize Salesforce Platform Event Notifications (or Messages) With ServiceNow Incidents	4-27
Insert or Update Records Using the Upsert Operation	4-29

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
Troubleshoot the Platform Event Service in the Salesforce Adapter	5-3
Error When the Upsert Operation is Performed Without Mapping the External ID Field	5-3

Preface

This guide describes how to configure this adapter as a connection in an integration in Oracle Integration.

**Note:**

The use of this adapter may differ depending on the features you have, or whether your instance was provisioned using Standard or Enterprise edition. These differences are noted throughout this guide.

Topics:

- [Audience](#)
- [Documentation Accessibility](#)
- [Diversity and Inclusion](#)
- [Related Resources](#)
- [Conventions](#)

Audience

This guide is intended for developers who want to use this 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.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our

initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

Related Resources

See these Oracle resources:

- Oracle Cloud
<http://cloud.oracle.com>
- *Using Integrations in Oracle Integration Generation 2*
- *Using the Oracle Mapper with Oracle Integration Generation 2*

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.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
<code>monospace</code>	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

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](#)
- [Salesforce Adapter Restrictions](#)
- [What Application Version Is Supported?](#)
- [Salesforce Adapter Use Cases](#)
- [Workflow to Create and Add a Salesforce Adapter Connection to an Integration](#)



Note:

There are overall service limits with Oracle Integration. A service limit is the quota or allowance set on a resource. See [Service Limits](#).

Salesforce Adapter Capabilities

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

The Salesforce Adapter provides the following benefits:

- Enables simplified bidirectional integration with [Salesforce.com](#).
- Discovers business objects and operations and provides easy mapping to and from [Salesforce.com](#) business objects.
- Provides ease of use in the mapper by recommending the most frequently-used business objects and renders human readable names for the elements found in Salesforce business objects for easy mapping.
- Provides support for performing the following type of operations against business objects fetched from the Salesforce application:
 - Bulk Create, Update, Upsert, Delete, and so on operations on all of the business objects supported by Salesforce
 - Core (Convertlead, GetDeleted, GetUpdated, Process, Merge, Undelete, and Upsert)
 - CRUD (create, delete, retrieve, and update) operations
 - Declaratively defining and executing SOQL and SOSL queries
 - Salesforce Object Search Language (SOSL) search operation
 - Utility (GetUserInfo, GetServerTimestamp, and SendEmail) operations

See [Process Large Data Sets Asynchronously with Different Bulk Operations](#) and [Use Bulk Response Operations in an Integration](#).

- Supports all custom objects defined by the user and custom fields created at [Salesforce.com](#) along with the standard objects and fields.
- Supports consuming custom Apex classes developed and exposed as SOAP services in [force.com](#).
- Supports receiving notifications (events) sent with outbound messaging and the Streaming API (that is, platform events and change data capture events) from [Salesforce.com](#).
- Supports trigger (source) callbacks.
- Supports Salesforce APIs protected using OAuth two-legged and three-legged token-based authentication and user name token-based authentication.

 [Video](#)

 [Video](#)

 [Video](#)

Salesforce Adapter Restrictions

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 Salesforce Adapter supports only Bulk API version 1.0.
 - Client applications must adhere to Salesforce's SOAP API support policy and backward compatibility terms.
 - Not all the push topic queries are supported by Salesforce.
 - The platform events feature has the following restriction:
 - The maximum number of delivered event notifications within a 24-hour period, shared by all CometD clients:
 - * 50,000 for Performance and Unlimited Editions
 - * 25,000 for Enterprise Edition
 - * 10,000 for Developer Edition
- See [Platform Event Allocations](#).
- SOQL queries do not support the `FIELDS()` function to fetch data from Salesforce.com. Instead, explicitly specify all the fields needed in the SOQL query to get the data from accounts.
 - You cannot use the Salesforce Adapter to publish events to Salesforce. However, you can use the REST Adapter to satisfy this requirement.
 - The Salesforce Adapter does not support the PushTopic and Generic Events.

What Application Version Is Supported?

For information about which application version is supported by this adapter, see the Connectivity Certification Matrix.

See [Connectivity Certification Matrix](#).

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.

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.	Create a Salesforce Adapter Connection
2	Create the integration. When you do this, you add trigger and invoke connections to the integration.	Create Integrations and Add the Salesforce Adapter Connection to an Integration
3	Map data between the trigger connection data structure and the invoke connection data structure.	Map Data of <i>Using Integrations in Oracle Integration Generation 2</i>
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).	Manage Lookups of <i>Using Integrations in Oracle Integration Generation 2</i>
5	Activate the integration.	Manage Integrations of <i>Using Integrations in Oracle Integration Generation 2</i>
6	Monitor the integration on the dashboard.	Monitor Integrations of <i>Using Integrations in Oracle Integration Generation 2</i>

Step	Description	More Information
7	Track payload fields in messages during runtime.	Assign Business Identifiers for Tracking Fields in Messages and Manage Business Identifiers for Tracking Fields in Messages of <i>Using Integrations in Oracle Integration Generation 2</i>
8	Manage errors at the integration level, connection level, or specific integration instance level.	Manage Errors of <i>Using Integrations in Oracle Integration Generation 2</i>

2

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.

- [Prerequisites for Oracle Integration Release 20.2.2.0.0 \(200524.0200.35760\) or Later](#)
- [Prerequisite for Oracle Integration Releases Earlier Than 20.2.2.0.0](#)
- [Prerequisites to Use the Salesforce Username Password Security Policy](#)
- [Prerequisites to Use the Salesforce OAuth Three-Legged \(Authorization Code Credentials\) or Resource Owner Password Credentials Security Policy](#)
- [Prerequisites to Use Platform Events](#)
- [Prerequisite to Use Change Data Capture \(CDC\) Events](#)
- [Prerequisite to Use Outbound Messaging](#)
- [Prerequisite to Use Apex Web Services](#)

Prerequisites for Oracle Integration Release 20.2.2.0.0 (200524.0200.35760) or Later

When you create or edit your Salesforce Adapter connection on release 20.2.2.0.0 (200524.0200.35760) or later of Oracle Integration, you must specify the following details on the Connections page. Therefore, you must obtain the following details before creating or editing a connection.

Topics:

- Instance type. See [Identify the Instance Type of Your Salesforce Organization](#).
- Custom domain name (optional). If you are using Salesforce Government Cloud or a custom domain to log in to your Salesforce account, a custom domain name is required for creating a connection on the Connections page. See [Obtain the Custom Domain Name](#).

- API version. See [Find Your Current Salesforce API Version](#).

Identify the Instance Type of Your Salesforce Organization

Follow these methods to identify your instance type:

- If you use the URL <https://login.salesforce.com> to log in to your Salesforce account, your instance type is **Production**.
- If you use the URL <https://test.salesforce.com> to log in to your Salesforce account, your instance type is **Sandbox**.

Obtain the Custom Domain Name

When you create your Salesforce Adapter connection on the Connections page, you must specify your custom domain name.

Note:

The steps provided here apply to the Salesforce Classic UI. If you are using the Lightning Experience UI on your Salesforce instance, switch to the Salesforce Classic UI. See [Toggle or switch between Lightning Experience and Salesforce Classic](#).

To obtain the custom domain name:

1. Log in to the Salesforce sandbox or production instance.
2. Click **Setup** in the header or click your username, then select **Setup**. See [Explore the Salesforce Setup Menu](#).
3. Search for `My Domain` in the **Quick Find** box, then select **My Domain**.
4. Under **My Domain Settings**, your custom domain name appears in the **Your domain name is** field.
See [Configure Connection Properties](#).

Find Your Current Salesforce API Version

You can find your current Salesforce API version in the enterprise WSDL.

Note:

The steps provided here apply to the Salesforce Classic UI. If you're using the Lightning Experience UI on your Salesforce instance, switch to the Salesforce Classic UI. See [Toggle or switch between Lightning Experience and Salesforce Classic](#).

To find your current Salesforce API version.

1. Log in to the Salesforce sandbox or production instance.

2. Click **Setup** in the header or click your username, then select **Setup**. See [Explore the Salesforce Setup Menu](#).
3. Search for `API` in the **Quick Find** box, then select **API** under **Develop**.
4. Click **Generate Enterprise WSDL**.
5. On the Generate Enterprise WSDL page, click **Generate**.

The WSDL opens in a new browser tab and your current API version appears in the Commented-out section. For example:

```
Salesforce.com Enterprise Web Services API Version 35.0
```

Prerequisite for Oracle Integration Releases Earlier Than 20.2.2.0.0

Perform the following prerequisite for Oracle Integration releases earlier Than 20.2.2.0.0.

Generate the Salesforce Enterprise WSDL

If you edit an existing connection created prior to release 20.2.2.0.0 (200524.0200.35760) of Oracle Integration, you must upload an enterprise WSDL to use in an integration.

You must generate the Salesforce enterprise WSDL, then upload this WSDL on the Connections page when creating a Salesforce Adapter connection.



Note:

The steps provided here apply to the Salesforce Classic UI. If you're using the Lightning Experience UI on your Salesforce instance, switch to the Salesforce Classic UI. See [Toggle or switch between Lightning Experience and Salesforce Classic](#).

To generate the Salesforce Adapter enterprise WSDL:

1. Log in to your Enterprise, Unlimited, or Developer Edition [Salesforce.com](#) account.
2. Click **Setup** in the header or click your username, then select **Setup**. See [Explore the Salesforce Setup Menu](#).
3. Search for `API` in the **Quick Find** box, then select **API** under **Develop**.
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, Microsoft Edge/Google Chrome users can select **Save link as**, while Mozilla Firefox users can select **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](#).

Prerequisites to Use the Salesforce Username Password Security Policy

To use the Salesforce Username Password security policy, you must create a user account in `Salesforce.com`.

1. Create an API-enabled profile. You assign this profile to the user account you subsequently create for Oracle Integration. See [Create a Secure Salesforce API User](#).
2. Create a user account in Salesforce for Oracle Integration. See [Add a Single User](#).
Once the user account is created, a verification email is sent to the email address you provided for the account.
3. Log in to the corresponding email account and click **Verify Account** in the email message from Salesforce.
You are redirected to the Salesforce instance to set a password for the new user account.

 **Note:**

You use the credentials of this user account when configuring the Salesforce connection in Oracle Integration.

4. Set a password and copy this value.
You're subsequently signed in to the Salesforce instance with the new account.
You'll need to enter these values on the Connections page when you configure the Salesforce Username Password security policy for your Salesforce Adapter connection in Oracle Integration. See [Configure Connection Security](#).
5. Generate a security token.
Generate a security token for the new user account. You'll need this security token along with the password to access Salesforce using Oracle Integration.
Stay signed in as the new user, and click the user name at the top of the page to open a menu.

 **Note:**

The steps provided here apply to the Salesforce Classic UI. If you're using the Lightning Experience UI on your Salesforce instance, switch to the Salesforce Classic UI. See [Toggle or switch between Lightning Experience and Salesforce Classic](#).

- a. Click **My Settings** in the menu or click the username, then select **My Settings**.
- b. On the My Settings page, in the **Quick Links** section, click **Edit my personal information**.

- c. On the resulting page, click **Reset My Security Token** in the left navigation pane.
- d. Click **Reset Security Token**.
A new security token is sent to the email address associated with the account.
- e. Copy the security token.

Prerequisites to Use the Salesforce OAuth Three-Legged (Authorization Code Credentials) 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](https://salesforce.com).



Note:

The steps provided here apply to the Salesforce Classic UI. If you are using the Lightning Experience UI on your Salesforce instance, switch to the Salesforce Classic UI. See [Toggle or switch between Lightning Experience and Salesforce Classic](#).

1. Log in to your Enterprise, Unlimited, or Developer Edition [Salesforce.com](https://salesforce.com) account.
2. Click **Setup** in the header or click your username, then select **Setup**. See [Explore the Salesforce Setup Menu](#).
3. Search for **Apps** in the **Quick Find** box, then select **Apps** under **Create**.
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](#).

Prerequisites to Use Platform Events

Perform the following prerequisites to use platform events.

Topics:

- [Define Your Platform Event](#)
- [Publish Platform Events](#)

Define Your Platform Event

To receive notifications using platform events, you must define the platform event. The list of platform events you defined in [Salesforce.com](#) is displayed for use on the Streaming API page of the Adapter Endpoint Configuration Wizard when configuring the Salesforce Adapter as a trigger connection.

 **Note:**

The steps provided here apply to the Salesforce Classic UI. If you are using the Lightning Experience UI on your Salesforce instance, switch to the Salesforce Classic UI. See [Toggle or switch between Lightning Experience and Salesforce Classic](#).

1. Log in to your Enterprise, Unlimited, or Developer Edition [Salesforce.com](#) account.

2. Click **Setup** in the header or click your username, then select **Setup**. See [Explore the Salesforce Setup Menu](#).
3. Search for Platform Events in the **Quick Find** box, then select **Platform Events**.
4. Click **New Platform Event**.
5. Enter the required details, and click **Save**.
6. To add a field, click **New** in the **Custom Fields & Relationships** section.
7. Select the field type, and click **Next**.
8. Enter the required details, and click **Save**. A new custom field is added.
Additional information about platform events is provided. See [Define and Manage Platform Events](#).

Publish Platform Events

After a platform event is defined, you must publish event messages from a Salesforce application.



Note:

See [Publishing Platform Events](#).

Prerequisite to Use Change Data Capture (CDC) Events

Perform the following prerequisite to use change data capture (CDC) events.

Select an Entity (Business Object)

To receive notifications using change data capture (CDC) events, you must select the entity/entities. Once selected, these objects are displayed for use on the Operations page of the Adapter Endpoint Configuration Wizard when configuring the Salesforce Adapter as a trigger connection.



Note:

The steps provided here apply to the Salesforce Classic UI. If you are using the Lightning Experience UI on your Salesforce instance, switch to the Salesforce Classic UI. See [Toggle or switch between Lightning Experience and Salesforce Classic](#).

1. Log in to your Enterprise, Unlimited, or Developer Edition [Salesforce.com](#) account.
2. Click **Setup** in the header or click your username, then select **Setup**. See [Explore the Salesforce Setup Menu](#).
3. Search for Change Data Capture in the **Quick Find** box, then select **Change Data Capture**.
4. Select the required entities (business objects) for which you want to receive change event notifications.

5. Click **Save**.

Prerequisite to Use Outbound Messaging

To use outbound messaging, perform the following prerequisites.



Note:

New workflow rules are blocked in the Salesforce application as of the Winter 2023 release. New workflow rules cannot be created in Salesforce. However, you can activate, deactivate, and edit existing workflow rules.

- [Use Outbound Messaging with a Flow](#)
- [Use Outbound Messaging with a Workflow](#)

Use Outbound Messaging with a Flow

To use outbound messaging with a flow, follow these instructions.

- [Create the Salesforce Outbound Message](#)
- [Develop a Flow in Salesforce](#)



Note:

- You must first create an outbound message and then develop a flow.
- These steps apply to the Salesforce Classic UI. If you are using the Lightning Experience UI on your Salesforce instance, switch to the Salesforce Classic UI. See [Toggle or switch between Lightning Experience and Salesforce Classic](#).

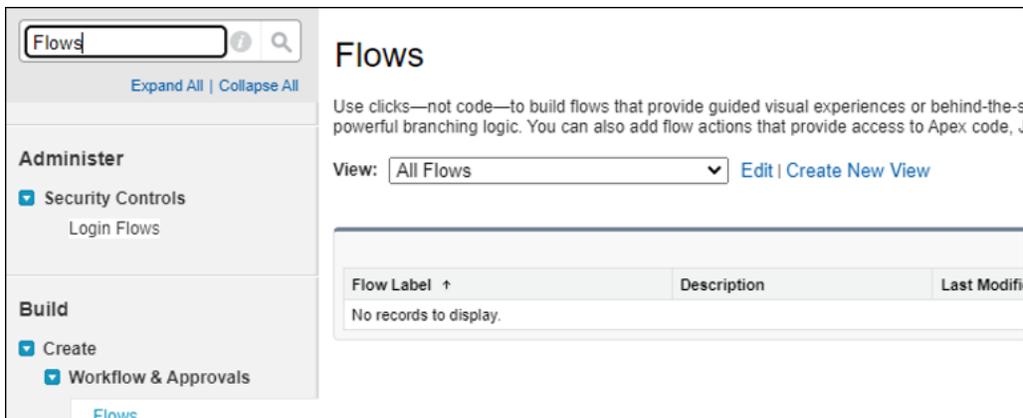
Create the Salesforce Outbound Message

1. Log in to your Salesforce account.
2. Click **Setup** in the header or click your user name, then select **Setup**. See [Explore the Salesforce Setup Menu](#).
3. Search for `Outbound Messages` in the **Quick Find** box.
4. Click **Outbound Messages** under **Workflow & Approvals** in the **Build** section.
5. Select an object.
6. Enter the required details.
7. Click **Save**.

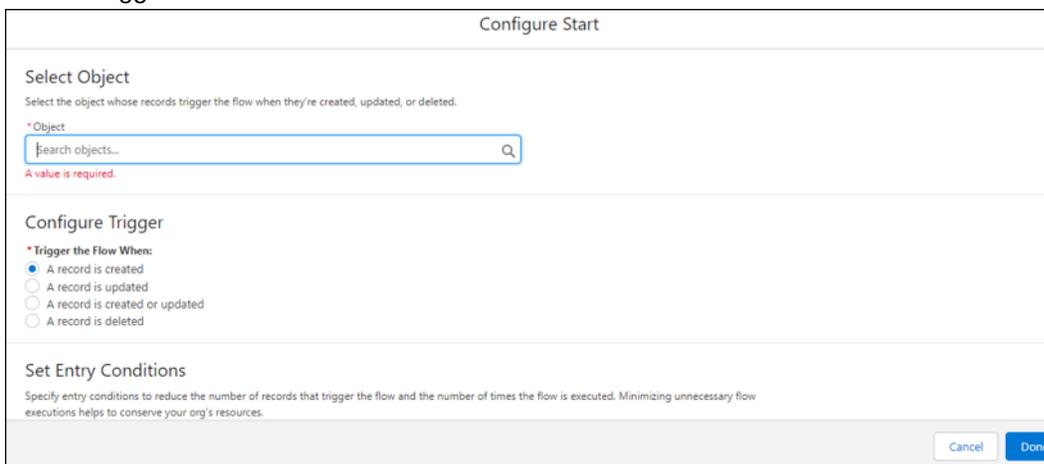
Develop a Flow in Salesforce

1. Log in to your Salesforce account.

2. Click **Setup** in the header or click your user name, then select **Setup**. See [Explore the Salesforce Setup Menu](#).
3. Search for **Flows** in the **Quick Find** box.
4. Select **Flows** under **Workflow & Approvals** in the **Build** section.



5. Click **New Flow**.
6. Select **Record-Triggered Flow**.
7. Click **Create**. The Configure Start page appears.
8. Select an object.
9. Set the trigger criteria.



10. Select the condition requirement in the **Condition Requirements** field. You click **add condition** to add an element.

Configure Start

Condition Requirements

All Conditions Are Met (AND) New Formulas for Conditions

Field: Search fields... Operator: Select... Value: Enter value or search resources...

+ Add Condition

When to Run the Flow for Updated Records

Every time a record is updated and meets the condition requirements

Only when a record is updated to meet the condition requirements

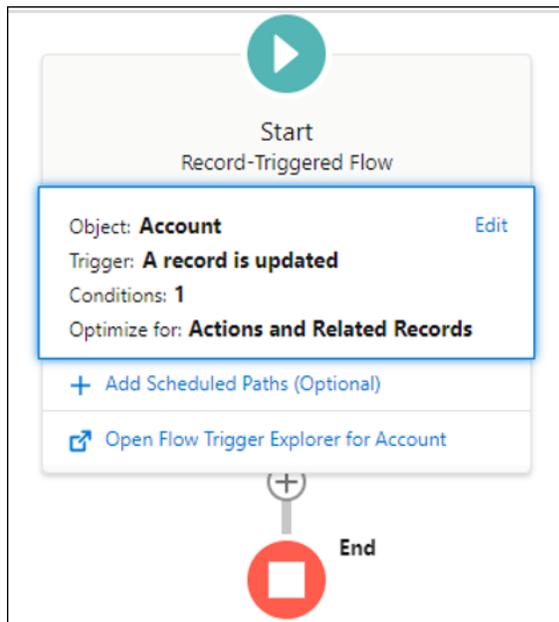
*Optimize the Flow for:

Fast Field Updates: Update fields on the record that triggers the flow to run. This high-performance flow runs before the record is saved.

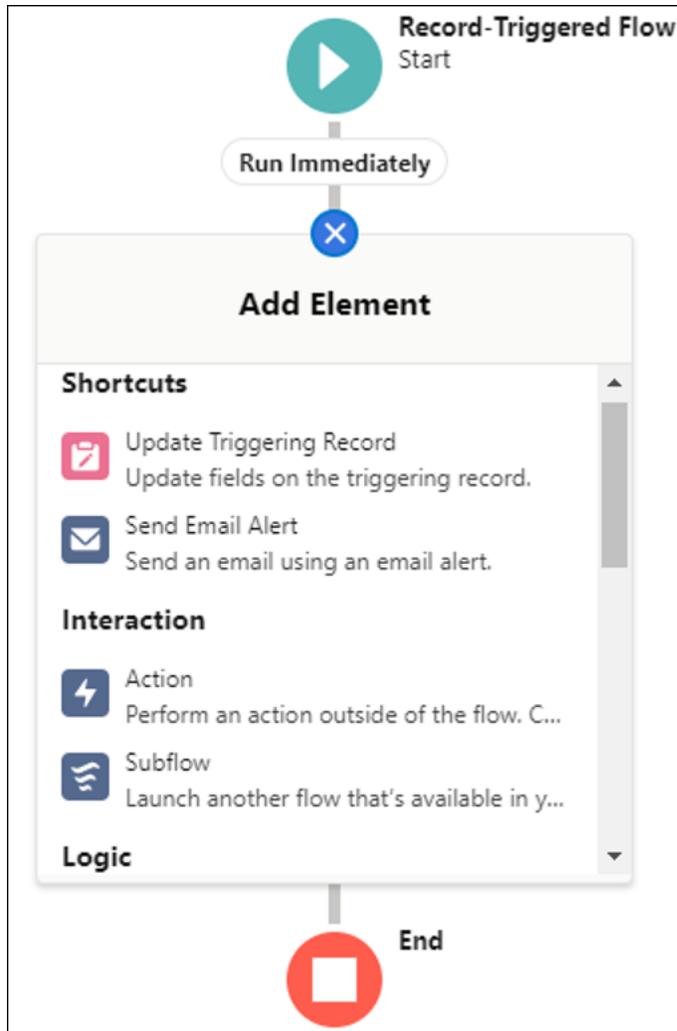
Actions and Related Records: Update any record and perform actions, like send an email. This more flexible flow runs after the record is saved to the

Cancel Done

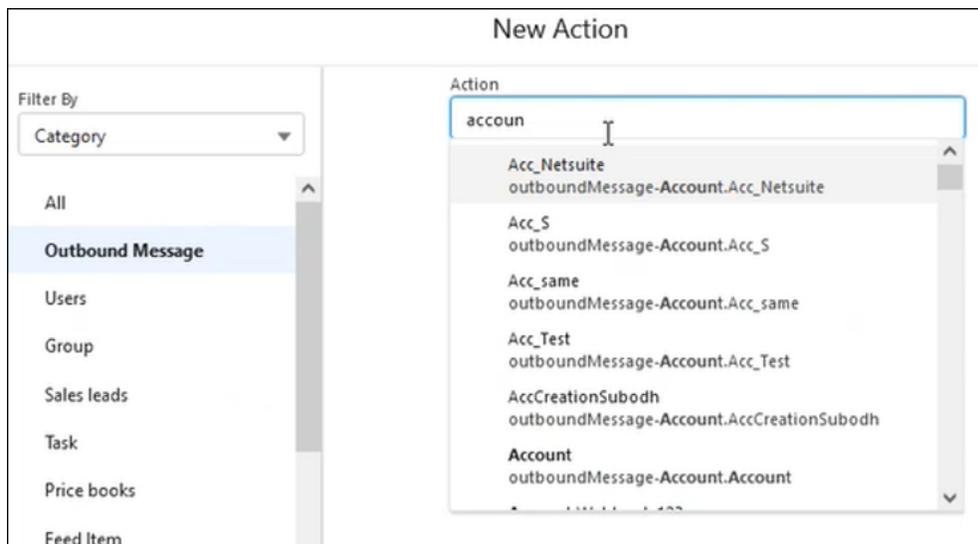
11. Select **Actions and Related Records**.
12. Click **Done**.
13. Click the **+** icon.



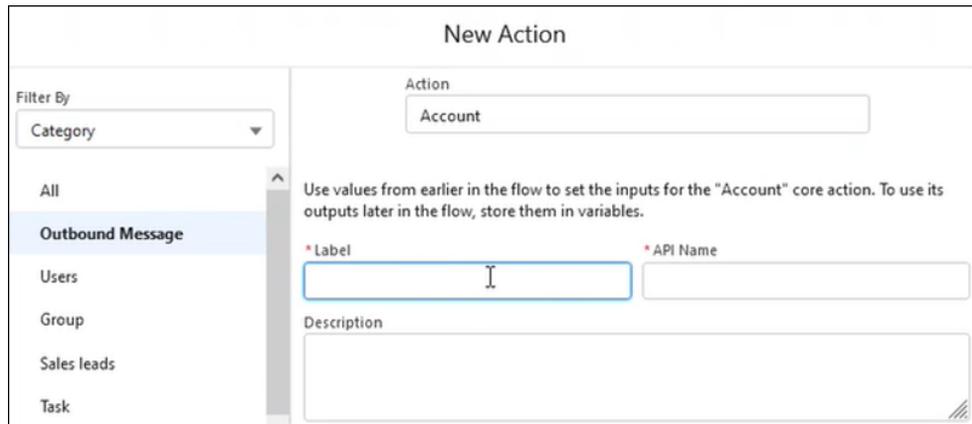
14. Select **Action** from **Add Element**.



15. Click **Outbound message**.
16. Select the outbound message that you created.



17. Enter a label name of the outbound message.



18. Click **Done** and click **Save**.
19. Enter a flow name and click **Save**.
20. Click **Save** and click **Activate**.

Use Outbound Messaging with a Workflow

Perform the following prerequisite to use outbound messaging.

Generate the Salesforce Adapter Outbound Messaging WSDL

You can generate an outbound messaging WSDL for the Salesforce Adapter to receive outbound message notifications from the Salesforce application. You then select this WSDL when configuring the Salesforce Adapter as a trigger on the Outbound Messaging page of the Adapter Endpoint Configuration Wizard.

This process consists of several steps. You must first create a workflow rule and generate an outbound messaging WSDL. 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.

Note:

The steps provided here apply to the Salesforce Classic UI. If you are using the Lightning Experience UI on your Salesforce instance, switch to the Salesforce Classic UI. See [Toggle or switch between Lightning Experience and Salesforce Classic](#).

To create a workflow rule:

1. Log in to your Salesforce account.
2. Click **Setup** in the header or click your username, then select **Setup**. See [Explore the Salesforce Setup Menu](#).

3. Search `Workflow` in the **Quick Find** box.
4. Click **Workflow Rules** under **Workflow & Approvals** in the **Build** section.
5. Click **New Rule**.
6. Select an object and click **Next**.
7. Enter a rule name, select an evaluation and rule criteria, and click **Save & Next**.
8. Click **Add Workflow Action** and select **New Outbound Message** or **Select Existing Action** (if you already have the outbound message).
9. Provide the required information in the wizards and click **Save**.
 - a. Enter a name, description for this outbound message, and endpoint URL, and select the fields you want included in the outbound message.

 **Note:**

You can enter a dummy URL in the **Endpoint URL** field. For example:

```
https://test.com
```

- b. Click **Add**.

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

1. Generate 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. See [Trigger Outbound Messaging Page](#).

 **Note:**

The steps provided here apply to the Salesforce Classic UI. If you are using the Lightning Experience UI on your Salesforce instance, switch to the Salesforce Classic UI. See [Toggle or switch between Lightning Experience and Salesforce Classic](#).

To generate the Salesforce outbound messaging WSDL:

1. Log in to your Salesforce account.
2. Click **Setup** in the header or click your username, then select **Setup**. See [Explore the Salesforce Setup Menu](#).
3. Search `Outbound Messages` in the **Quick Find** box.
4. Click **Outbound Messages** under **Workflow & Approvals** in the **Build** section.
5. Select the required outbound message.
6. Click **Click for WSDL** to download the outbound messaging WSDL.
7. Drag the Salesforce Adapter to the trigger (inbound) section of the integration canvas. This invokes the Adapter Endpoint Configuration Wizard.

8. Browse for the generated WSDL on the Outbound Messaging page.
9. Activate the integration and copy the endpoint URL from the integration information icon.
10. Select the required outbound message.
11. Click **Edit**.
12. Replace the dummy URL with the activated integration endpoint URL in the **Endpoint URL** field. You entered the dummy URL when configuring your outbound message.

Prerequisite to Use Apex Web Services

Perform the following prerequisite to use Apex web services.

Generate the Salesforce Adapter Custom WSDL

You can generate 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 select the custom WSDL for the **Invoke Apex Web Services** option when configuring the Salesforce Adapter in the invoke direction. See [Invoke Apex Operations Page](#).

For more information about custom WSDLs, see [Exposing Apex Methods as SOAP Web Services](#).

Create a Connection

Before you can build an integration, you have to create the connections to the applications with which you want to share data.

To create a connection in Oracle Integration:

1. In the left navigation pane, click **Home > Integrations > 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.)

3. In the Create Connection — Select Adapter dialog, select the adapter to use for this connection. To find the adapter, scroll through the list, or enter a partial or full name in the **Search** field and click



Search.

4. In the Create Connection dialog, enter the information that describes this connection.
 - a. 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, don't include blank spaces (for example, SALES OPPORTUNITY).
 - b. Enter optional keywords (tags). You can search on the connection keywords on the Connections page.
 - c. Select the role (direction) in which to use this connection (trigger, invoke, or both). Only the roles supported by the 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, you'll get an error when you try to drag the adapter into the section you didn't select. For example, let's say you configure a connection for the Oracle Service Cloud (RightNow) Adapter as only an **invoke**. Dragging the adapter to a **trigger** section in the integration produces an error.
 - d. Enter an optional description of the connection.
5. Click **Create**.

Your connection is created. You're now ready to configure the connection details, such as connection properties, security policies, connection login credentials, and (for certain connections) agent group.

Configure Connection Properties

Enter connection information so your application can process requests.

1. Go to the **Connection Properties** section.

The fields that are displayed are based on your instance version.
2. Specify details based on your instance version.

 **Note:**

If you create or edit your Salesforce Adapter connection on Oracle Integration release 21.2.1.0.0 (211110.1100.42360) or later, you must specify API version 47 or later in the **API Version** field or upload an enterprise WSDL of API version 47 or later.

To Edit Existing or Create New Connections on...	Description
Release 20.2.2.0.0 (200524.0200.35760) or later of Oracle Integration	<p>a. From the Select Salesforce.com Instance Type list, select the instance type:</p> <ul style="list-style-type: none"> • Production • Sandbox • Government: Select if you want to connect to Salesforce Government Cloud. <p>Note: For more information about instance type, see Identify Instance Type of Your Salesforce Organization.</p> <p>b. In the Custom domain field, enter your custom domain name. For example, if your custom domain name is <code>sfcustomdomain.my.salesforce.com</code>, then enter <code>sfcustomdomain</code>. See Obtain the Custom Domain Name.</p> <p>Note:</p> <ul style="list-style-type: none"> • The Government instance type option and Custom domain field are displayed for connections created in release 20.3.3 or later of Oracle Integration. • The Custom domain field is optional. However, if you are using Salesforce Government Cloud or a custom domain to log in to your Salesforce account, the custom domain name is required. • The Salesforce Adapter recommends that you <i>not</i> use your Salesforce instance value as your custom domain name (for example, with the URL <code>https://apl.salesforce.com</code>, the value before <code>salesforce.com</code> is your Salesforce instance value). • Don't use common words like <code>test</code>, <code>login</code>, <code>sfcustomdomain.my</code>, and <code>sfcustomdomain.my.salesforce.com</code> as the custom domain name. <p>c. In the API Version field, enter the API version to use. See Find Your Current Salesforce API Version.</p> <p>Note: If you edit an existing connection created on release 20.2.2.0.0 (200524.0200.35760) or later of Oracle Integration, the instance type and API version fields are displayed.</p>

To Edit Existing or Create New Connections on...	Description
A release prior to Oracle Integration 20.2.2.0.0 (200524.0200.35760)	<p>a. 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 Generate the Salesforce Enterprise WSDL.</p> <p>Note: If you edit an existing connection created prior to release 20.2.2.0.0 (200524.0200.35760) of Oracle Integration, the field to upload an enterprise WSDL is displayed.</p>

Configure Connection Security

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

1. Go to the **Security** section.
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 `SSSSSSSSSS` and the password is `password`, you must enter `passwordSSSSSSSSSS` 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 [Prerequisites to Use the Salesforce OAuth Three-Legged \(Authorization Code Credentials\) 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 [Prerequisites to Use the Salesforce OAuth Three-Legged \(Authorization Code Credentials\) 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 [Prerequisites to Use the Salesforce OAuth Three-Legged \(Authorization Code Credentials\) 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 [Prerequisites to Use the Salesforce OAuth Three-Legged \(Authorization Code Credentials\) 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 [Prerequisites to Use the Salesforce OAuth Three-Legged \(Authorization Code Credentials\) 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 (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 `SSSSSSSSSSSS` 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.

- 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's configured successfully.

1. In the page title bar, click **Test**. What happens next depends on whether your connection uses a Web Services Description Language (WSDL) file.

If Your Connection...	Then...
Doesn't use a WSDL	The test starts automatically and validates the inputs you provided for the connection.
Uses a WSDL	A dialog prompts you to select the type of connection testing to perform: <ul style="list-style-type: none"> • 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.

2. Wait for a message about the results of the connection test.
 - If the test was successful, then the connection is configured properly.
 - If the test failed, then edit the configuration details you entered. Check for typos, verify URLs and credentials, and download the diagnostic logs for additional details. Continue to test until the connection is successful.
3. When complete, click **Save**.

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 left navigation pane, click **Home > Settings > Certificates**. All certificates currently uploaded to the trust store are displayed in the Certificates dialog. The



link enables you to filter by name, certificate expiration date, status, type, category, and installation method (user-installed or system-installed). Certificates installed by the system cannot be deleted.

Certificates			
Name	Type	Category	Status
mykey3 <small>EXPIRES IN 1 MONTH</small>	X.509	Identity	Configured
mykey2 <small>EXPIRES</small>	X.509	Identity	Configured
recert1586867745048 <small>EXPIRES IN 4 YEARS</small>	X.509	Trust	Configured
recert1586863610817 <small>EXPIRES IN 4 YEARS</small>	X.509	Trust	Configured
recert1586857607511 <small>EXPIRES IN 4 YEARS</small>	X.509	Trust	Configured
recert1586857416600 <small>EXPIRES IN 4 YEARS</small>	X.509	Trust	Configured

- Click **Upload** at the top of the page. The Upload Certificate dialog box is displayed.
- Enter an alias name and optional description.
- In the **Type** field, select the certificate type. Each certificate type enables Oracle Integration to connect with external services.
 - X.509 (SSL transport)**
 - SAML (Authentication & Authorization)**
 - PGP (Encryption & Decryption)**

X.509 (SSL transport)

- Select a certificate category.
 - Trust:** Use this option to upload a trust certificate.
 - Click **Browse**, then select the trust file (for example, `.cer` or `.crt`) to upload.
 - Identity:** Use this option to upload a certificate for two-way SSL communication.
 - Click **Browse**, then select the keystore file (`.jks`) to upload.
 - Enter the comma-separated list of passwords corresponding to key aliases.

Note:

When an identity certificate file (JKS) contains more than one private key, all the private keys must have the same password. If the private keys are protected with different passwords, the private keys cannot be extracted from the keystore.

- Enter the password of the keystore being imported.
- Click **Upload**.

SAML (Authentication & Authorization)

- Note that **Message Protection** is automatically selected as the only available certificate category and cannot be deselected. Use this option to upload a keystore

certificate with SAML token support. Create, read, update, and delete (CRUD) operations are supported with this type of certificate.

2. Click **Browse**, then select the certificate file (.cer or .crt) to upload.
3. Click **Upload**.

PGP (Encryption & Decryption)

1. Select a certificate category. Pretty Good Privacy (PGP) provides cryptographic privacy and authentication for communication. PGP is used for signing, encrypting, and decrypting files. You can select the private key to use for encryption or decryption when configuring the stage file action.
 - a. **Private:** Uses a private key of the target location to decrypt the file.
 - i. Click **Browse**, then select the PGP file to upload.
 - ii. Enter the PGP private key password.
 - b. **Public:** Uses a public key of the target location to encrypt the file.
 - i. Click **Browse**, then select the PGP file to upload.
 - ii. In the **ASCII-Armor Encryption Format** field, select **Yes** or **No**. **Yes** shows the format of the encrypted message in ASCII armor. ASCII armor is a binary-to-textual encoding converter. ASCII armor formats encrypted messaging in ASCII. This enables messages to be sent in a standard messaging format. This selection impacts the visibility of message content. **No** causes the message to be sent in binary format.
 - iii. From the **Cipher Algorithm** list, select the algorithm to use. Symmetric-key algorithms for cryptography use the same cryptographic keys for both encryption of plain text and decryption of cipher text.
 - c. Click **Upload**.

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 left navigation pane, click **Home > Integrations > Connections**.
2. Go to the row of the connection to refresh.

3. Select **Refresh Metadata** from the  menu.

A message is displayed indicating that the refresh was successful.

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

3

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 Action Page](#)
- [Trigger Outbound Messaging Page](#)
- [Trigger Streaming API Page](#)
- [Trigger Response Page](#)
- [Trigger Callback Response 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. The Salesforce Cloud Basic Info page is the initial wizard page that is displayed whenever you drag an adapter to the trigger or invoke area.

Element	Description
What do you want to call your endpoint?	<p>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 <code>SalesforceOutboundDirection</code>. You can include English alphabetic characters, numbers, underscores, and dashes in the name. You cannot include the following:</p> <ul style="list-style-type: none">• Blank spaces (for example, <code>My Salesforce Connection</code>)• Special characters (for example, <code>#;83&</code> or <code>right(now4)</code>)• Multibyte characters

Element	Description
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.

Trigger Action Page

Select the following to receive notifications from the Salesforce application.

Element	Description
How would you like to receive messages from Salesforce.com	<ul style="list-style-type: none">• Outbound Messaging: Select to receive outbound message notifications from the Salesforce application.• Streaming API: Select to receive event notifications from the Salesforce application using platform events and change data capture events. You receive event notifications for the platform events you defined in Salesforce.com and receive change event notifications for the entities you selected in Salesforce.com.

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 [Generate the Salesforce Adapter Outbound Messaging WSDL](#).

If you selected **Outbound Messaging** on the Action Page, the following option is displayed.

Element	Description
Select the Outbound Messaging WSDL	<p>Select the invoke Salesforce outbound messaging WSDL to receive outbound message notifications from the Salesforce application.</p> <p>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.</p> <ol style="list-style-type: none">1. Browse for and select the invoke Salesforce outbound messaging WSDL (.wsdl file). The Select New Outbound Messaging WSDL field is displayed. <i>This is an optional field.</i>2. If you want to optionally upload a new or modified Salesforce outbound messaging WSDL to replace the already-uploaded outbound messaging WSDL, then select that WSDL (.wsdl file).3. Activate the integration and copy the endpoint URL from the integration information icon.4. Go to the Outbound Messaging section at www.salesforce.com and replace the dummy URL you previously entered with the real endpoint URL.

Trigger Streaming API Page

Select the platform events and change data capture events to use with the Salesforce Adapter.

If you selected **Streaming API** on the Action Page, the following options are displayed. Select the option appropriate to your integration requirements in the **Select a streaming event type** field.

- [Subscribe to platform events published on Salesforce.com](#)
Select to receive event notifications from the Salesforce application based on the events you defined in Salesforce.com.

You must have already defined the platform events in Salesforce.com. See [Define Your Platform Event](#).

- [Subscribe to change data capture events published on Salesforce.com](#)
Select to receive change event notifications, whenever a data change in Salesforce occurs, for the entities (business objects) you selected in Salesforce.com

You must have already selected the change data capture events in Salesforce.com. See [Select an Entity \(Business Object\)](#).

 **Note:**

The Salesforce Adapter does not support the Overflow event.

Subscribe to platform events published on Salesforce.com

Element	Description
Select a platform event	<p>Displays the list of platform events you defined in Salesforce.com. Select the platform event to receive notifications from the Salesforce application.</p> <p>Note:</p> <ul style="list-style-type: none"> For a new integration, the Salesforce Adapter receives event notifications from the last 72 hours for the selected platform event channel because the subscription occurs with <code>replayId</code> set to -2. For an existing integration, which was inactive for the last 72 hours or more and then activated again, it receives only new platform events for the subscribed channel because the subscription occurs with <code>replayId</code> set to -1.
Filter by: *<KEYWORD>*	Type a keyword or the initial letters of the event to filter the display of names in the list.

An implementation pattern using platform events is provided. See [Synchronize Salesforce Platform Event Notifications \(or Messages\) With ServiceNow Incidents](#).

Subscribe to change data capture events published on Salesforce.com

Element	Description
Select a change event type	<p>Select one or both options:</p> <ul style="list-style-type: none"> Normal: Select to receive the common change events notifications whenever a data change in Salesforce occurs. Gap: Select this option if you want to receive gap event notifications. Gap events are generated instead of change events in special situations. See Gap Events.
Select a change type	<ul style="list-style-type: none"> Create: Select to receive notifications whenever a new record is created under the selected entity. Update: Select to receive notifications whenever a record is updated under the selected entity. Delete: Select to receive notifications whenever a record is deleted under the selected entity. Undelete: Select to receive notifications whenever a record is undeleted under the selected entity.
Select a business object	Displays the list of entities (business objects) you selected in Salesforce.com. Select the business object to receive change event notifications whenever a data change in Salesforce occurs for the selected entity.

Element	Description
Filter by: *<KEYWORD>*	Type a keyword or the initial letters of the event to filter the display of names in the list.

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.

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.

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.

Element	Description
Select an Operation Type	<p>Select the type of operation to perform on the business objects in a Salesforce Cloud application:</p> <ul style="list-style-type: none"> • 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. <p>Note: For the Update operation, external ID cannot be mapped to ID.</p> <ul style="list-style-type: none"> • 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	<p>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:</p> <ul style="list-style-type: none"> • 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)	<p>Select a single business object or multiple business objects from the Salesforce Cloud application. The selected operation acts upon these business objects.</p> <p>When you complete invoke operation configuration, the selected operation and business objects are defined in the integration-centric WSDL file.</p>
Your Selected Business Objects	Displays the business objects you selected.

Invoke Action Page

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

Element	Description
What would you like to do in Salesforce.com?	<ul style="list-style-type: none">• 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 or REST 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

Element	Description
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.	<ul style="list-style-type: none"><li data-bbox="881 333 1383 474">• 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. Use the queryMore operation in the following scenarios:<ul style="list-style-type: none"><li data-bbox="930 552 1383 604">– If you need to fetch more than 2000 records from Salesforce.com.<li data-bbox="930 615 1383 863">– If you enter an SOQL query that selects more than two custom fields of type long text and it does not return a required set of records with the query operation. A use case is provided for the queryMore operation. See Retrieve the Name and IDs from Account Records with the queryMore Operation. <p data-bbox="930 873 1276 900">Deleted and achieved records:</p> <ul style="list-style-type: none"><li data-bbox="930 911 1383 1052">– 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.<li data-bbox="930 1062 1383 1226">– 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. <ul style="list-style-type: none"><li data-bbox="881 1236 1383 1289">• Retrieve: Retrieves accounts from Salesforce application.<li data-bbox="881 1299 1383 1434">• Search: Returns records from the Salesforce application. You can specify binding parameters to dynamically provide a search string as input to your search operation.

Element	Description
Enter a Salesforce Object Query Language (SOQL) Statement	<p>Enter a valid query statement. SOQL statements evaluate to a list of sObjects, a single sObject, or an integer for <code>count</code> method queries. The following examples are provided:</p> <pre data-bbox="878 426 1378 514">"SELECT Id FROM Contact WHERE Name LIKE 'A%' AND MailingCity = 'California'"</pre> <pre data-bbox="878 583 1378 615">SELECT COUNT() FROM Contact</pre> <p>SOSL statements evaluate to a list of sObjects, where each list contains the search results for a particular sObject type. For example:</p> <pre data-bbox="878 814 1378 913">"SELECT a.name, a.id, a.accountNumber, c.name from Contact c, c.Account"</pre>
Click here for SOQL Query Reference Binding Parameters	<p>Contains a list of SOQL query samples. Displays any parameters included in the query. For example, <code>orgId</code> is a parameter in the following query:</p> <pre data-bbox="878 1129 1378 1249">SELECT a.name, a.id, a.accountNumber, c.name from Contact c, c.Account a WHERE a.name = "&orgId"</pre>
Test My Query	<p>This query displays a binding parameters text box in which to enter a test value for <code>orgId</code>.</p> <p>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.</p>

Create, Update Or Delete Information Operation Type

Element	Description
If you selected Create, Update Or Delete Information on the Action page, the following option is displayed.	<ul style="list-style-type: none">• Create, Update, Upsert, or Delete Information: Represents the create, update, upsert, or delete operation to perform on Salesforce business objects. A use case is provided for the upsert operation. See Insert or Update Records Using the Upsert Operation. <p>Note: For the Upsert operation, an external ID field must be mapped to a Salesforce API field name. Otherwise, you receive an error, see Error When the Upsert Operation is Performed Without Mapping the External ID Field.</p>

Perform Bulk Data Operations

Element	Description
If you selected Perform Bulk Data Operations on the Action page, the following options are displayed.	<ul style="list-style-type: none">• Create• Delete• FinalBatch• 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.• 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.• Update• Upsert <p>A use case is provided for this operation. See Process Large Data Sets Asynchronously with Different Bulk Operations.</p>

Perform Core or Utility Operations

Element	Description
<p>If you selected Perform Core or Utility Operations on the Action page, the following options are displayed.</p>	<ul style="list-style-type: none"> • Convert Lead: Converts a lead into an account, a contact, or an opportunity in Salesforce.com. • Get Deleted: Retrieves deleted IDs of the Salesforce record of a selected object for a limited time period (maximum of one month). • Get Server Timestamp: 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 dateTime that contains the server timestamp value. • Get Updated: Retrieves IDs of an updated Salesforce record of a selected object for a limited time period (maximum of one month). • Get User Info: 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 userId, userFullName, userEmail, profileId, roleId, organizationId, and so on. • Merge: This operation merges up to three records into one record in Salesforce.com. • Process: 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. <ul style="list-style-type: none"> – ProcessSubmitRequest – ProcessWorkitemRequest • Undelete: Restores Salesforce.com records from the recycle bin. • Send Email: Triggers a single email message request to Salesforce. This operation takes most standard email attributes, templateIds, targetObjectIds, and other attributes as input. This operation responds with the Boolean element Success that defines the status of the message triggered. If the email message is successfully triggered, it responds with true. Otherwise, it responds with false. <ul style="list-style-type: none"> – Send Mass Email: Triggers a mass email message request to Salesforce. This operation takes most standard email attributes, templateIds, multiple targetObjectIds, and other attributes as input. This operation returns the Boolean element Success that defines the status

Element	Description
	of the message triggered. If the email message is successfully triggered, it responds with true . Otherwise, it responds with false .
Filter By Object Name	Type the initial letters to filter the display of business objects. You can also select a filter type: <ul style="list-style-type: none"> • 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 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.

Element	Description
All Or None Header (request header)	<p>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.</p> <ul style="list-style-type: none">• Integration flows created before version 16.4.1:<ul style="list-style-type: none">– If set to true and the response contains error elements, it maps to <code>UnexpectedErrorFault</code>. This is also the case if All Or None Header is set to false (that is, unselected).• Integration flows created for version 16.4.1 and later:<ul style="list-style-type: none">– If set to true (that is, selected and there are error elements), it maps to <code>UnexpectedErrorFault</code>.– 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. <p>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.</p>
Allow Field Truncation Header (request header)	<p>Specifies the truncation behavior for the following fields (each are string data types):</p> <ul style="list-style-type: none">• <code>anyType</code>• <code>email</code>• <code>picklist</code>• <code>encryptedstring</code>• <code>textarea</code>• <code>multipicklist</code>• <code>phone</code>• <code>string</code> <p>Set Allow Field Truncation to one of the following values:</p> <ul style="list-style-type: none">• 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
Assignment Rule Header (request header)	<p>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 <code>AssignmentRule</code> object. If the ID is specified, you do not need to specify the Use Default Rule value.</p> <ul style="list-style-type: none">• Assignment Rule Id: 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.• Use Default Rule: If set to true, the default (active) assignment rule is used. If set to false, the default (active) assignment rule is not used.
Duplicate Rule Header	<p>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:</p> <ul style="list-style-type: none">• Allow Save: This property is selected by default when configuring a Salesforce endpoint.<ul style="list-style-type: none">– True (if selected): Creates/updates duplicate records. The selection of this property disables the selection of Include Record Details and Run As Current User.– False (if deselected): Prevents created and updated duplicate records from being saved.• Include Record Details: This option is selectable when Allow Save is deselected. This option enables you to get record details of existing duplicates.<ul style="list-style-type: none">– True (if selected): Retrieves duplicate error details and duplicate record details that impact the transaction.– False(if deselected): Retrieves duplicate error details that do not include duplicate records.• Run As Current User: Enforces the sharing of rules for the current user.<ul style="list-style-type: none">– True (if selected): Runs duplicate rules for the current user and ignores the rules not available to them.– False (if deselected): Applies shared rules for the current user. <p>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.</p>

Element	Description
Email Header (request header)	<p>Specifies whether or not to send a notification email. You can set the following properties:</p> <ul style="list-style-type: none"> • Trigger Auto Response Email <ul style="list-style-type: none"> – true: Triggers automatic response rules for leads and cases. – false: Automatic response rules for leads and cases are not triggered. • Trigger Other Email <ul style="list-style-type: none"> – true: The email is triggered outside the organization. – false: The email is not triggered outside the organization. • Trigger User Email <ul style="list-style-type: none"> – 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)	<p>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:</p> <ul style="list-style-type: none"> • 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)	<p>Specifies the package version for any installed package. The package version identifies the components in a package. The package version follows the format <code>majorNumber.minorNumber.patchNumber</code> (for example, <code>3.4.5</code>, where 3 refers to <code>majorNumber</code>, 4 refers to <code>minorNumber</code>, and 5 refers to <code>patchNumber</code>).</p>
Query Options (request header)	<p>Specifies the batch size for queries. The minimum and default value is 200 and the maximum value is 2000.</p>

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

- [Apex SOAP](#)
- [Apex REST](#)

Apex SOAP

Apex SOAP provides support for consumption of SOAP endpoints exposed through Apex in the Salesforce Adapter.

Element	Description
Select the Apex WSDL	Click Choose File to select the Apex WSDL to use.
Operations in Uploaded WSDL	Displays the list of operations included in the uploaded Apex WSDL. Select the operation to perform in the Salesforce Cloud application.
Select New Apex WSDL	Click Choose File to select the Apex 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. You can use an Apex WSDL that includes Apex classes written on force.com and exposed as SOAP web services. See Exposing Apex Methods as SOAP Web Services .

Apex REST

Apex REST provides support for consumption of REST endpoints exposed through Apex in the Salesforce Adapter.

Note:

Configure the OAuth-based security policies for your Salesforce Adapter connection if you need to use the Apex REST operation. This is because the Apex REST operation is not supported when you configure the Username Password security policy for your Salesforce Adapter connection.

Element	Description
What is the endpoint's relative resource URI?	Specify the relative path associated with the resource. The path can contain template parameters specified with curly braces (for example, {order-id}). A resource is any source of specific information that can be addressed. The resource path follows a fixed, prefixed URL appended with the specified relative path. By default, the URL is prefixed with the following path: https:// yourInstance.salesforce.com/ services/apexrest/

Element	Description
What action do you want to perform on the endpoint?	<p>Select a single HTTP action (method) for the endpoint to perform:</p> <ul style="list-style-type: none"> • GET: Retrieves (reads) information (for example, makes queries). If you select this option, you cannot configure a request payload for this endpoint. • PUT: Updates information. • POST: Creates information. • DELETE: Deletes information. If you select this option, you cannot configure a request payload for this endpoint. • PATCH: Partially updates existing resources (for example, when you only need to update one attribute of the resource).
Select any options that you want to configure	<p>Based on your selections, you can add parameters or configure a request and/or response for this endpoint.</p> <ul style="list-style-type: none"> • Add and review parameters for this endpoint: Click to specify the query parameters and view the template request parameters created as part of the resource URI for this endpoint. If you select this option, the Query and Template Parameters page is displayed. • Configure a request payload for this endpoint: Click to configure the request payload for this endpoint, including specifying the JSON sample location. If you select this option, the Sample Request Payload page is displayed. Note: Only PUT and POST actions support this (Configure a request payload for this endpoint) option. Therefore, this option is displayed when you select the PUT or POST action. • Configure this endpoint to receive the response: Click to configure the response payload for this endpoint, including specifying the JSON sample location. If you select this option, the Sample Response Payload page is displayed.

Query and Template Parameters Page

If you select **Add and review parameters for this endpoint option** on the Apex Operations Page, the following options are displayed.

Element	Description
Specify Query Parameters	<p>Specify query parameters for the REST endpoint.</p> <p>Click the Add icon to display a row for entering the parameter name and selecting its data type. For example, specify <code>state</code> and select a data type of string.</p> <p>Click the Delete icon to delete a selected row.</p>
Template Parameters	<p>Displays the template parameters in the relative resource URI. Template parameters are based on details you specified on the Basic Info page and cannot be edited. Template parameters must be defined as part of a path with curly braces around them. For example, the URL <code>default/customers/{cust-id}/{ship-id}</code> has <code>cust-id</code> and <code>ship-id</code> template parameters.</p> <p>You can change the data type for the parameters.</p>

Invoke Sample Request Payload Page

If you select **Configure a request payload for this endpoint** on the Apex Operations page, the following options are displayed.

 **Note:**

The **Configure a request payload for this endpoint** option is visible only if the **PUT** or **POST** action is selected on the Apex Operations page.

Element	Description
Select the request payload format	<p>Select the request payload format to use.</p> <ul style="list-style-type: none"> JSON: Upload a JSON file containing a sample request payload or copy and paste the JSON payload into the text field. Binary: Use with payloads that are unstructured and inline (for example, <code>application/octet-stream</code>). This option preserves the file contents, but requires the receiver to determine the file type (for example, from the file name extension). The internet media type for an arbitrary byte stream is <code>application/octet-stream</code>. A list of commonly-used types is shown in a dropdown list. You can select a type from this list or provide a type not listed by selecting Other Media Type and entering the type in the text field. <p>Note: Ensure that the sample JSON is representative of the actual runtime messages exchanged with the endpoint. A mismatch in the structure or type of runtime messages can result in errors.</p>
Location of Sample Payload	<p>Specify the sample JSON request payload file in either of the following ways:</p> <ul style="list-style-type: none"> Click Choose File to select the sample JSON request payload to use. Click <<inline>> to copy and paste the JSON payload. Click Generate Element when complete.
Element	<p>Select the element that defines the payload structure. This field is not displayed until you import the request payload file.</p>
Select the type of payload with which you want the endpoint to send (If the Binary payload format is selected.)	<p>Select from a list of commonly-used types provided in the dropdown menu. You can also select Other Media Type to provide a type not listed in the dropdown list (for example, <code>video/mp4</code>).</p>

Invoke Sample Response Payload Page

If you select **Configure this endpoint to receive the response** on the Apex Operations page, the following options are displayed.

Element	Description
Select the response payload format	<p>The response payload can only be JSON.</p> <ul style="list-style-type: none"> Upload a JSON file containing a sample response payload. Copy and paste the JSON payload. <p>Note: Ensure that the sample JSON is representative of the actual runtime messages exchanged with the endpoint. A mismatch in the structure or type of runtime messages can result in errors.</p>

Element	Description
Location of Sample Payload	Specify the sample JSON response payload file in either of the following ways: <ul style="list-style-type: none"> Click Choose File to select the sample JSON response payload to use. Click <<inline>> to copy and paste the JSON payload. Click Generate Element when complete.
Element	Select the element that defines the payload structure. This field is not displayed until you import the response payload file.

Summary Page

You can review the specified adapter configuration values 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 . To cancel your configuration details, click Cancel .

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.

4

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](#)
- [Synchronize Salesforce Platform Event Notifications \(or Messages\) With ServiceNow Incidents](#)
- [Insert or Update Records Using the Upsert Operation](#)

Note:

Oracle Integration offers a number of prebuilt integrations, known as *recipes*, that provide you with a head start in building your integrations. You can start with a recipe, and then customize it to fit your needs and requirements. Depending upon the solution provided, a variety of adapters are configured in the prebuilt integrations.

See the Recipes and Accelerators page on the Oracle Help Center.

Process Large Data Sets Asynchronously with Different Bulk Operations

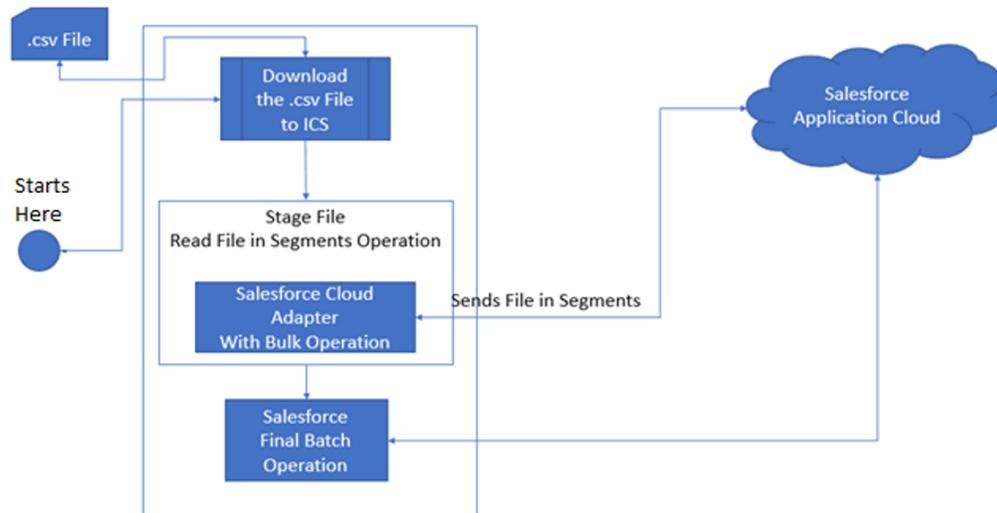
The Salesforce Bulk API enables you to handle huge data sets asynchronously with different bulk operations. For every bulk operation, the Salesforce application creates a job that is processed in batches.

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 8,000 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.

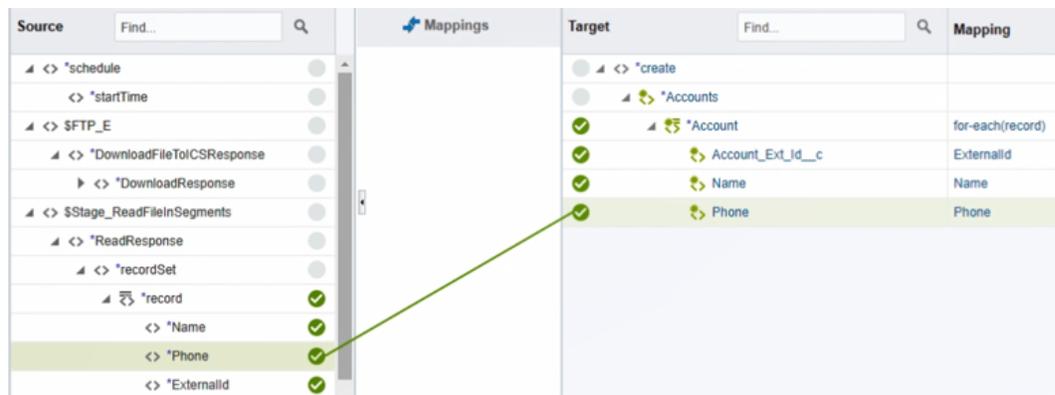
To perform this operation, you create FTP Adapter and Salesforce Adapter connections in Oracle Integration.

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.

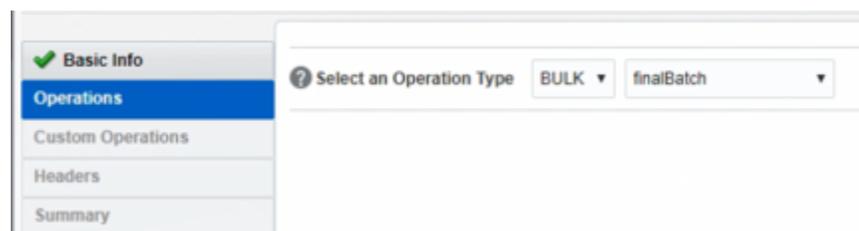
5. Drag a Salesforce Adapter inside the stage file action and configure it with the required **BULK** operation.
 - a. On the Basic Info page, provide a name.
 - b. Select **standard applications delivered by Salesforce.com**.
 - c. On the Action page, select **Perform Bulk Data Operations**.
 - d. On 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:**

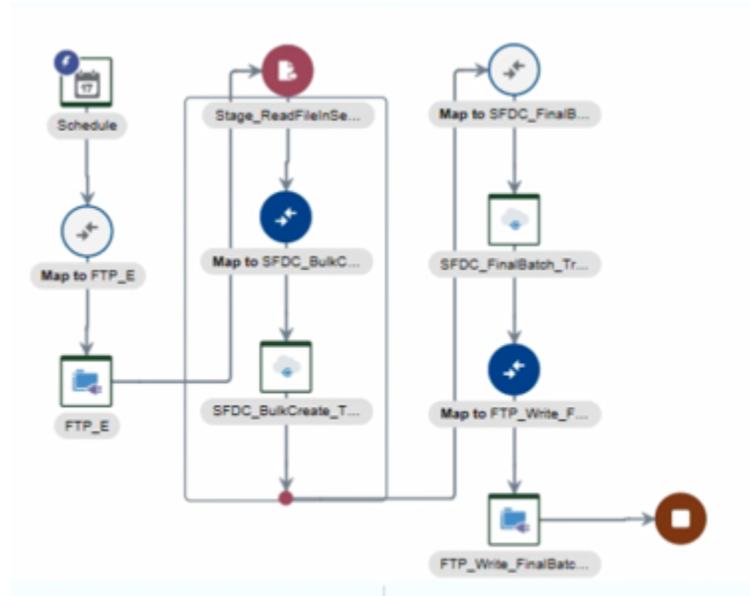
You cannot select the object on which to perform the operation because this Salesforce Adapter connection performs the activity for the operation that was configured for the Salesforce endpoint inside the stage file action.



- d. Review your selections on the Summary page.
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.
 - 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.

Source	Find...	Mappings	Target	Find...	Mapping
bulkResponse			*finalBatchResponse		
jobInfo			*bulkResponse		
id	✓		*jobInfo		
operation	✓		id	id	
object	✓		operation	operation	
createdById	✓		object	object	
createdDate	✓		createdById	createdById	
systemModstamp	✓		createdDate	createdDate	
state	✓		systemModstamp	systemModstamp	
concurrencyMode	✓		state	state	
contentType	✓		concurrencyMode	concurrencyMode	
numberBatchesQueued	✓		contentType	contentType	
numberBatchesInProgress	✓		numberBatchesQueued	numberBatchesQueued	

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).



11. 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.

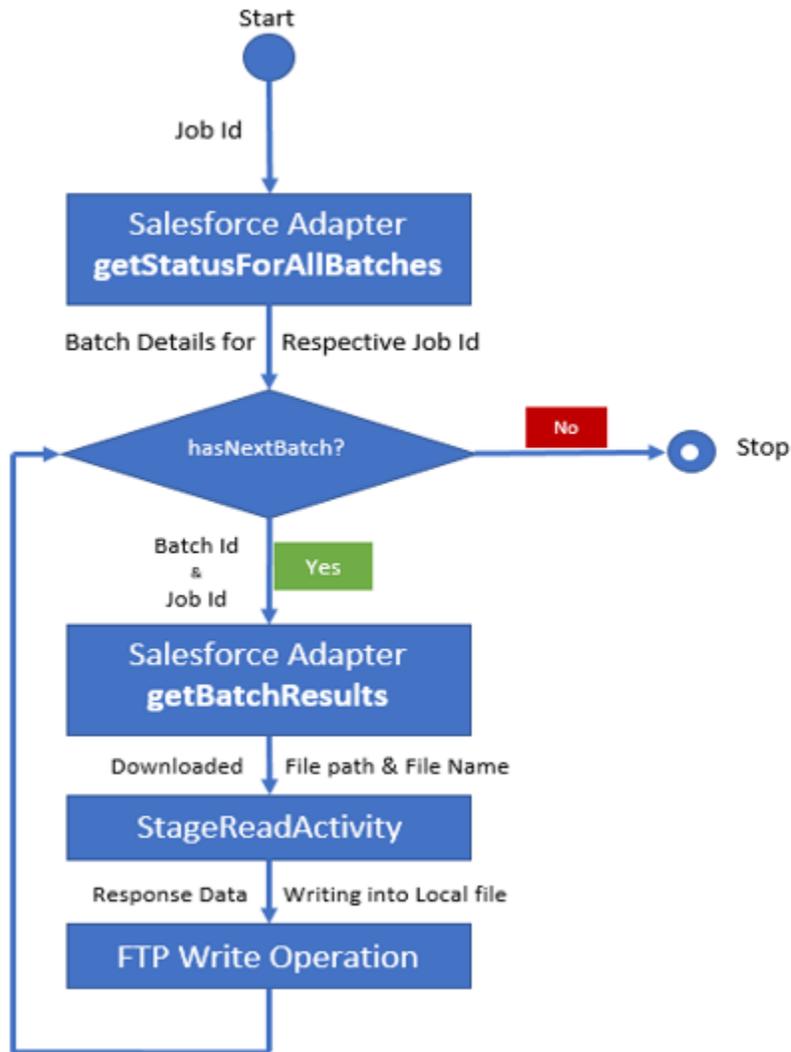
Use Bulk Response Operations in an Integration

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

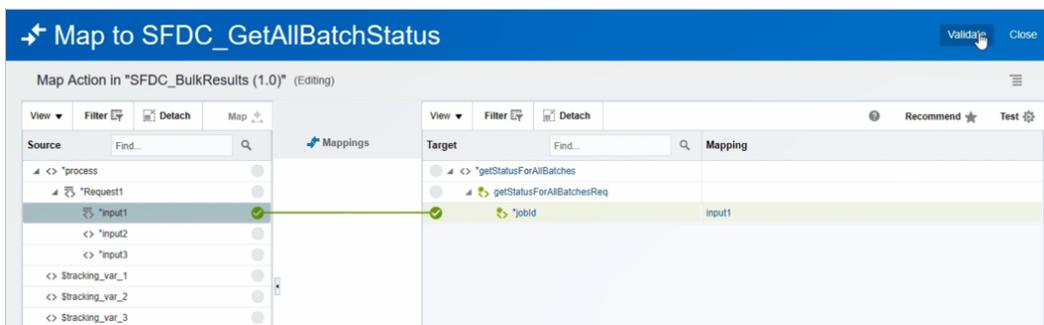


Note:

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

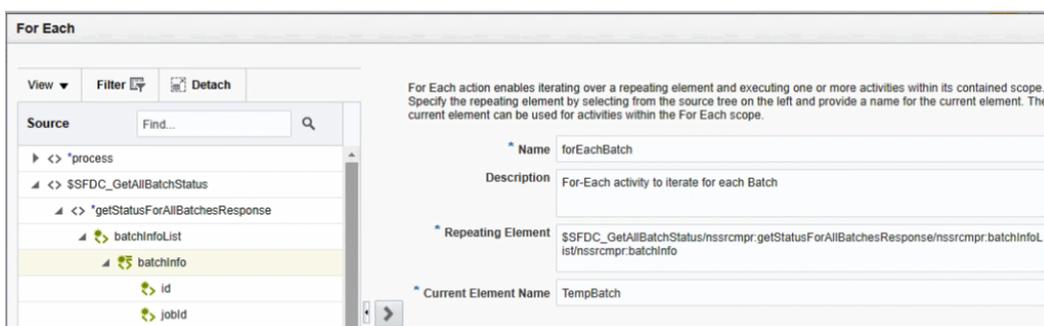
1. On the Integrations page, click **Create**.
2. Select the **Orchestration** pattern, click **Select**, then select **Application Event or business object** as the option to trigger this integration.
3. Drag the SOAP Adapter to the trigger section of the integration canvas.
4. Complete the Adapter 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.

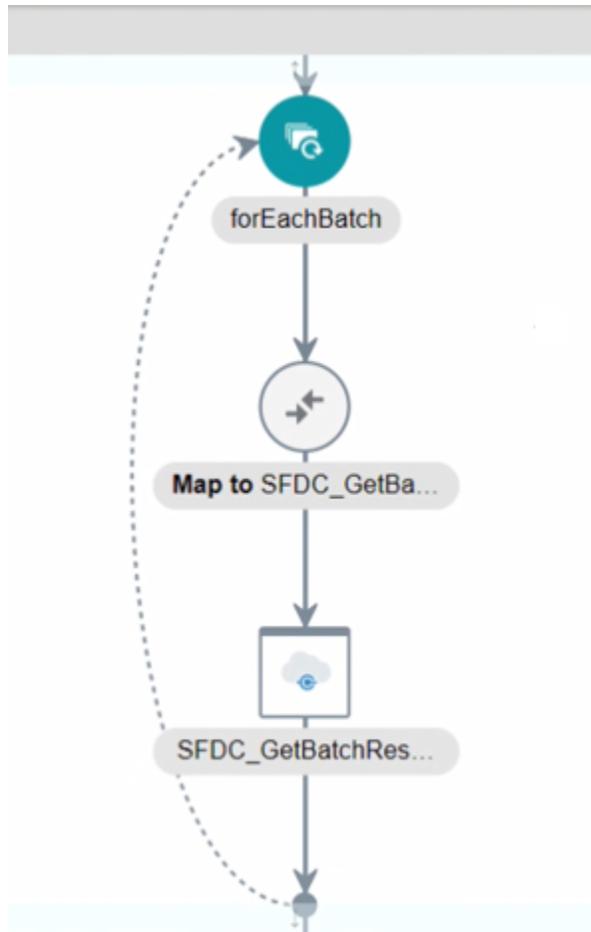


Create a For Each Action in the Integration

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**.



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**.

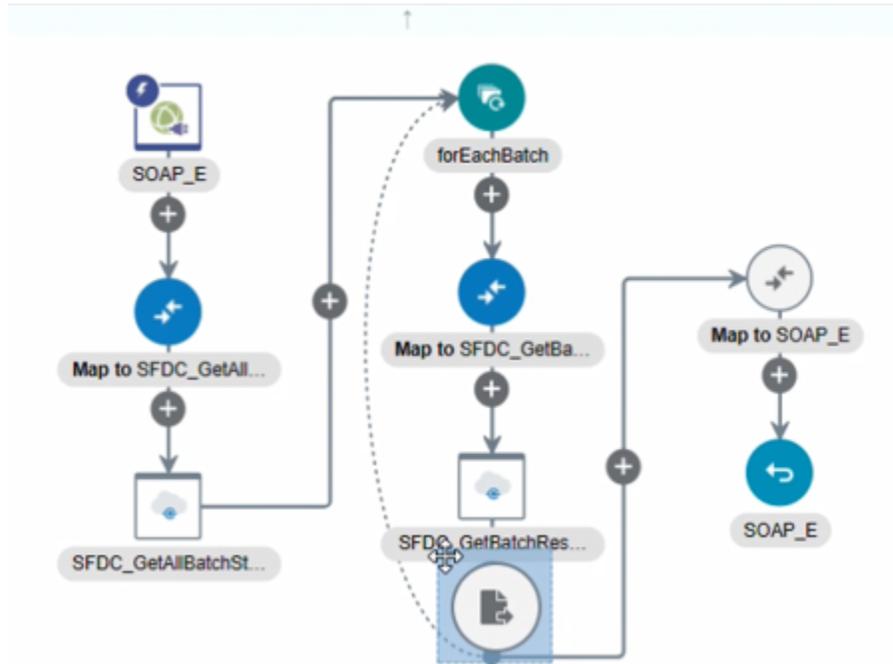
Map Action in "SFDC_BulkResults (1.0)" (Editing)

Source	Target	Mapping
<> *input2	*getBatchResults	
<> *input3	getBatchResultsReq	
▲ <> \$SFDC_GetAllBatchStatus	*jobId	jobId
▲ <> *getStatusForAllBatchesResponse	*batchId	id
▶ batchInfoList		
▲ <> \$TempBatch		
▲ *batchInfo		
▶ id		
▶ jobId		

7. Validate the mappings, then close the mapper and save your changes when prompted.

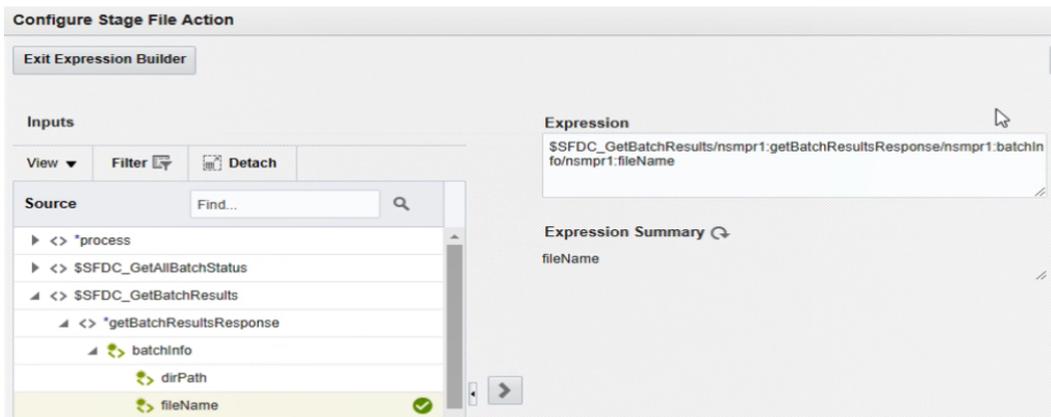
Add a Stage File Action to the For Each Action

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.

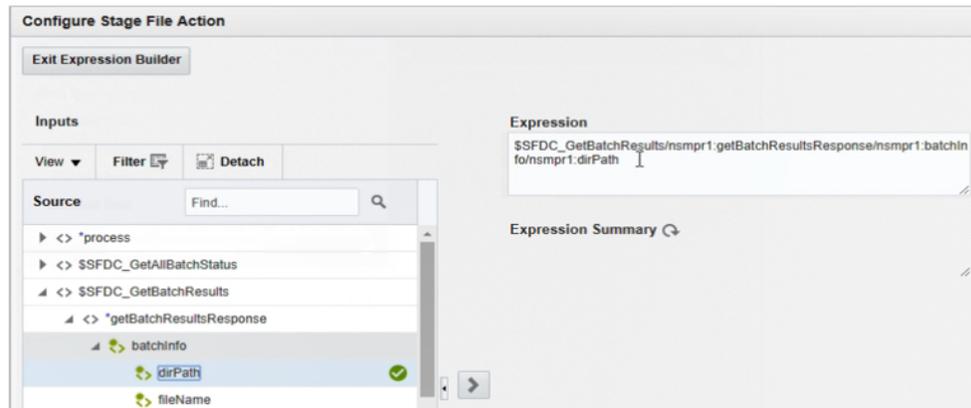


The Configure Stage File Action wizard is displayed.

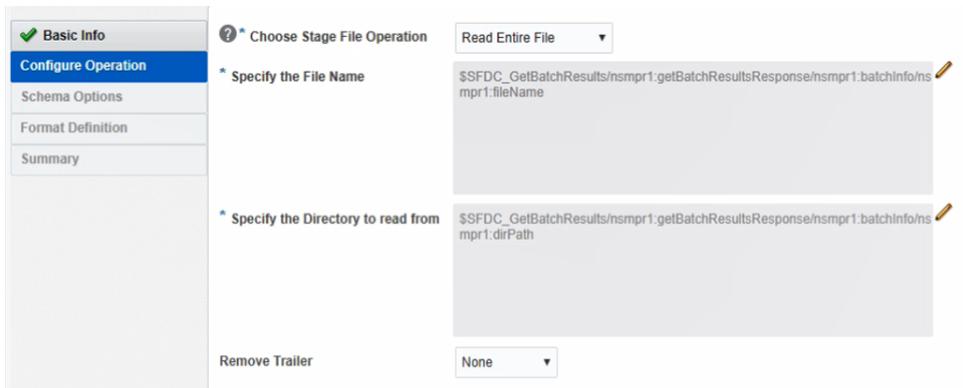
2. Provide a name, then click **Next**.
3. On the Configure Operation page, select **Read Entire File** from the **Choose Stage File Option** list, then click the **Select Expression** icon to specify the file name.
4. Drag the source **fileName** element to the **Expression** field, then click the **Expression 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**.

Create a New Schema from a CSV file

Select a New Delimited Data File No file chosen

Selected File Name result.csv

* Enter the Record Name

* Enter the Recordset Name

Select the Field Delimiter

Character Set

Optionally Enclosed By

Detach Use First Row as Column Headers Mark All As Optional

Id	Success	Created	Error
String	String	String	String
Mandatory	Mandatory	Mandatory	Mandatory
001e000000zw2Zu...	true	false	
001e000000zw2Zv...	true	false	

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.

Basic Info

Operations

Schema

Format Definition

Summary

* Select Operation

* Select a Transfer Mode ASCII Binary

* Specify an Output Directory

* Specify a File Name Pattern

Append to Existing File

* Enable PGP Security Yes No

4. On the Schema page, create a schema using a sample bulk file (for this example, response.csv).

Basic Info
 Operations
 Schema
 Format Definition
 Summary

? Do you want to define a schema for this endpoint?
 Yes No

? Do you want to create a new schema or select an existing one?
 Create a new schema from a CSV file
 Select an existing schema from the file system

- 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.

Create a New Schema from a CSV file

Select a New Delimited Data File No file chosen

Selected File Name result.csv

* Enter the Record Name

* Enter the Recordset Name

Select the Field Delimiter

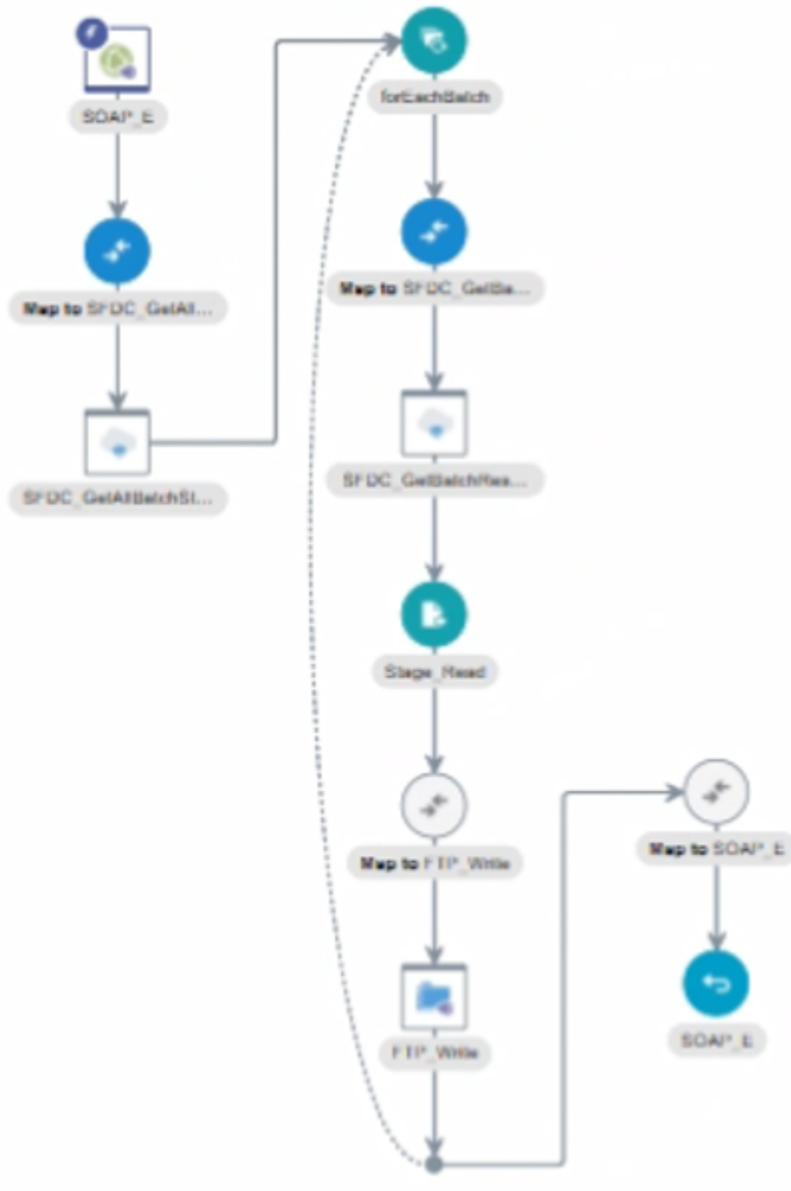
Character Set

Optionally Enclosed By

Detach Use First Row as Column Headers Mark All As Optional

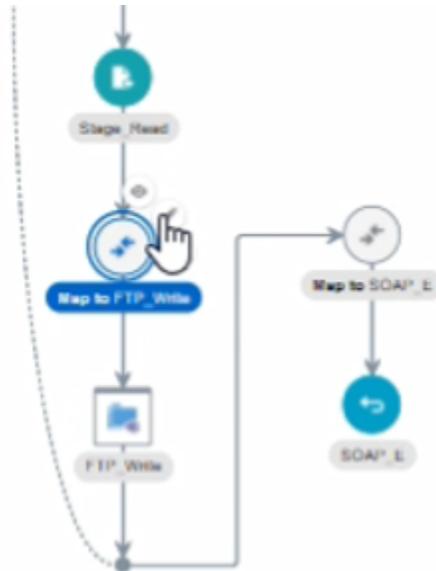
Id	Success	Created	Error
String	String	String	String
Mandatory	Mandatory	Mandatory	Mandatory
001e000000zw2Zu...	true	false	
001e000000zw2Zv...	true	false	

- 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.

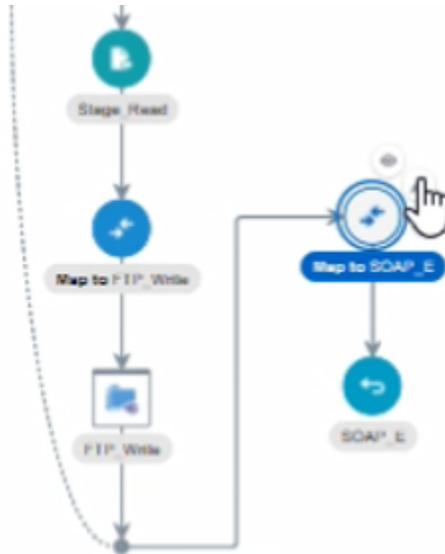


2. Map the stage response elements to the FTP request elements.

Map Action in "SFDC_BulkResults (1.0)" (Editing)

Source	Target	Mapping
<> *process	<> *WriteFile	
<> \$SFDC_GetAllBatchStatus	<> OutboundFTPHeaderType	
<> \$SFDC_GetBatchResults	<> fileName	
<> \$Stage_Read	<> directory	
<> *ReadResponse	<> *RecordSet	
<> *recordset	<> *Record	for-each(Record)
<> *Record	<> *id	id
<> *id	<> *Success	Success
<> *Success	<> *Created	Created
<> *Created	<> *Error	Error
<> *Error		
<> \$TempBatch		
<> \$tracking_var_1		
<> \$tracking_var_2		

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).

Map Action in "SFDC_BulkResults (1.0)" (Editing)

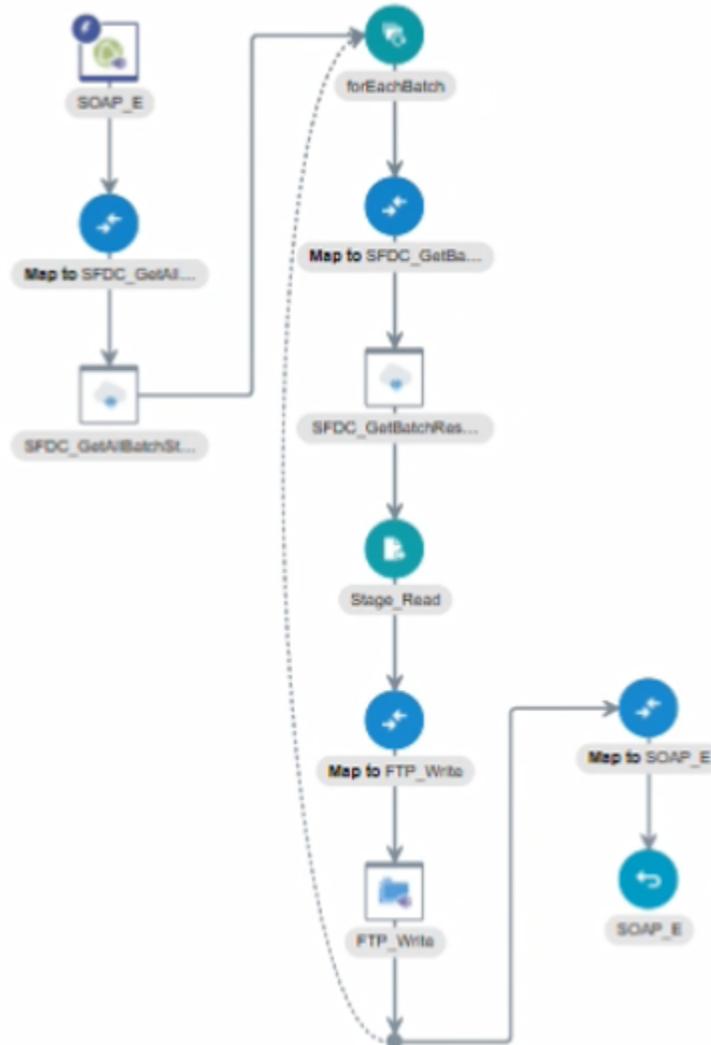
Source	Target	Mapping
<> Stracking_var_3	*processResponse	
<> SFDC_GetAllBatchStatus	*Response1	for-each(batchInfo)
*getStatusForAllBatchesResponse	<> *output1	id
batchinfoList	<> *output2	jobid
batchinfo	<> *output3	state
id	*error	
jobid	<> *ErrorMessage	
state	<> *Errorcode	

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

If you need to fetch more than 2000 records from Salesforce.com, use the **queryMore** operation. If you enter an SOQL query that selects more than two custom fields of type

long text and it does not return a required set of records with the query operation, you can use 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.

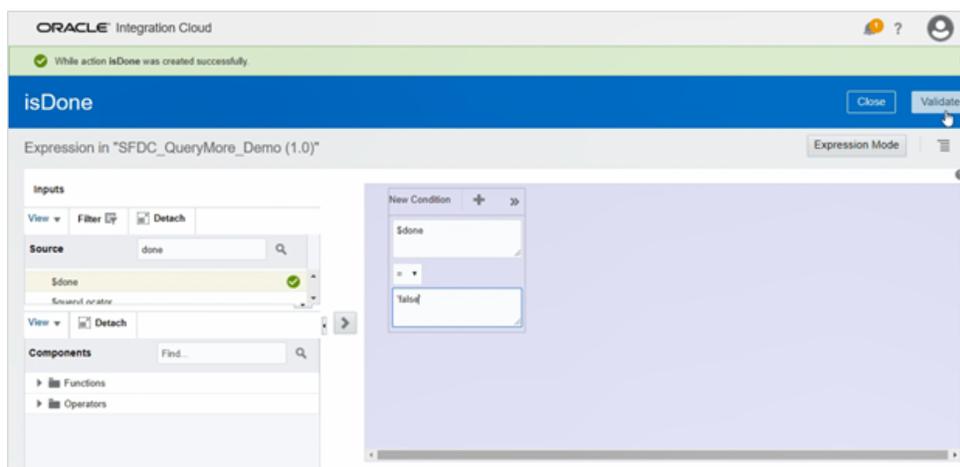
1. Create SOAP Adapter and Salesforce Adapter connections.
2. Create an app-driven orchestrated integration.
3. Drag the SOAP Adapter connection into the integration as a trigger connection.
4. Drag an assign action to the workspace next to the SOAP Adapter (for this example, named **Init**).

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**.

Variable	Data Type	Description	Operation	Value
(x) done	simple	Type a description		✎ "false"
(x) queryLocator	simple	Type a description		✎ ""
(x) countRecords	simple	Type a description		✎ 0.0

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
```

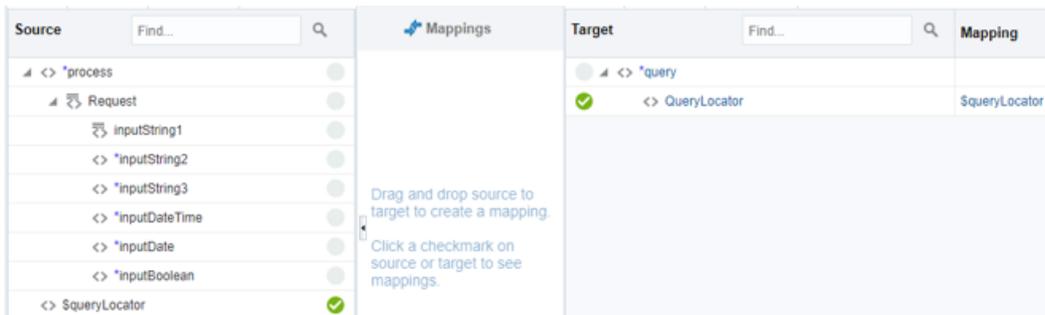
where `Select id, name, BillingCity` retrieves the ID, name, and billing city from the `Salesforce Account` object for all the records.

- 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.

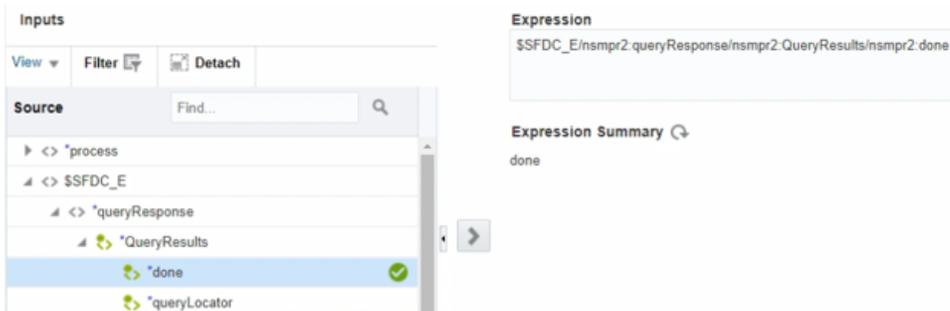
Note:

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.

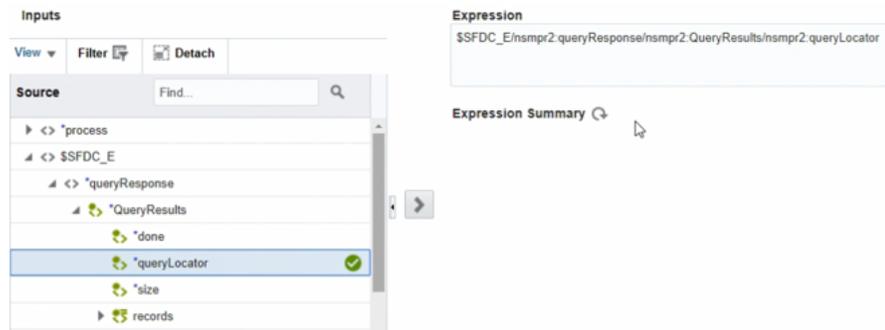
- In the mapper, map the local source **\$queryLocator** variable to the target **QueryLocator** element of the request mapper. For every iteration, the input for the Salesforce endpoint is the **queryLocator** value, which is initialized with "" (empty value) in the first configured assign action (named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



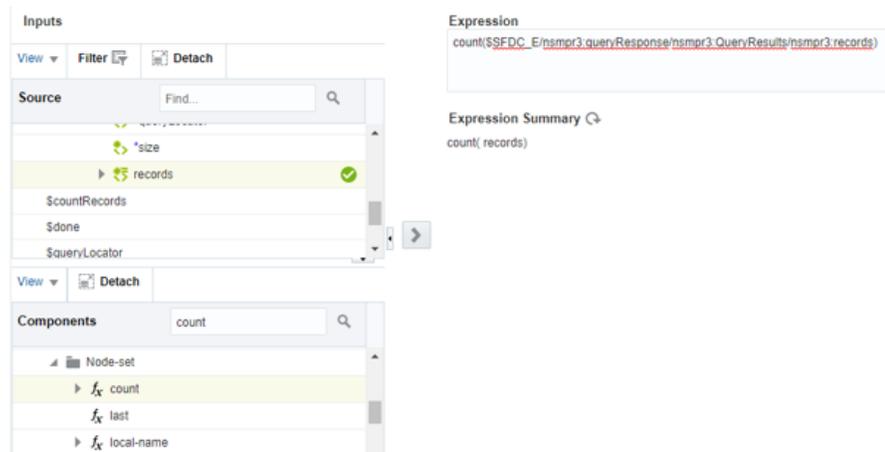
- Drag the assign action next to the Salesforce Adapter (for this example, named **Update**).
 - Update the local variables (**\$done** and **\$queryLocator**) with Salesforce output variables.
 - For the **\$done** variable:



- For the **\$queryLocator** variable:

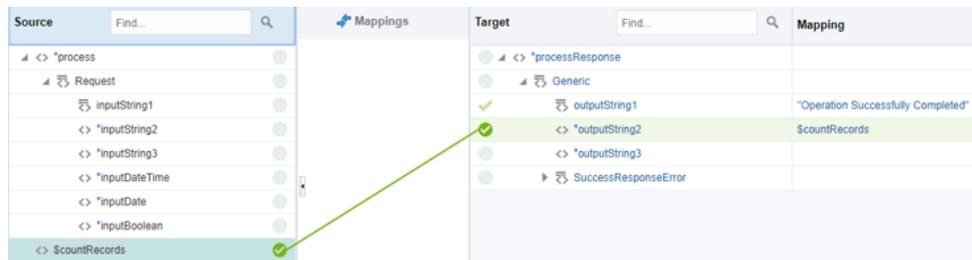


- For the **\$countRecords** variable:

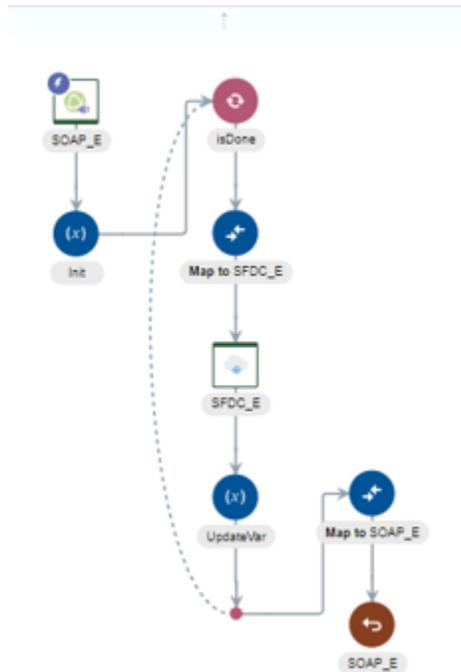


The **count()** function is available under **Functions**, categorized in **Node-set**. This takes **Node-set** as input and returns the count of nodes available in the **Node-set**. The record **Node-set** is passed to the **count ()**. After mapping, the function call looks as follows: **count(\$SFDC_E/nsmpr3:queryResponse/nsmpr3:QueryResults/nsmpr3:records)**

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.

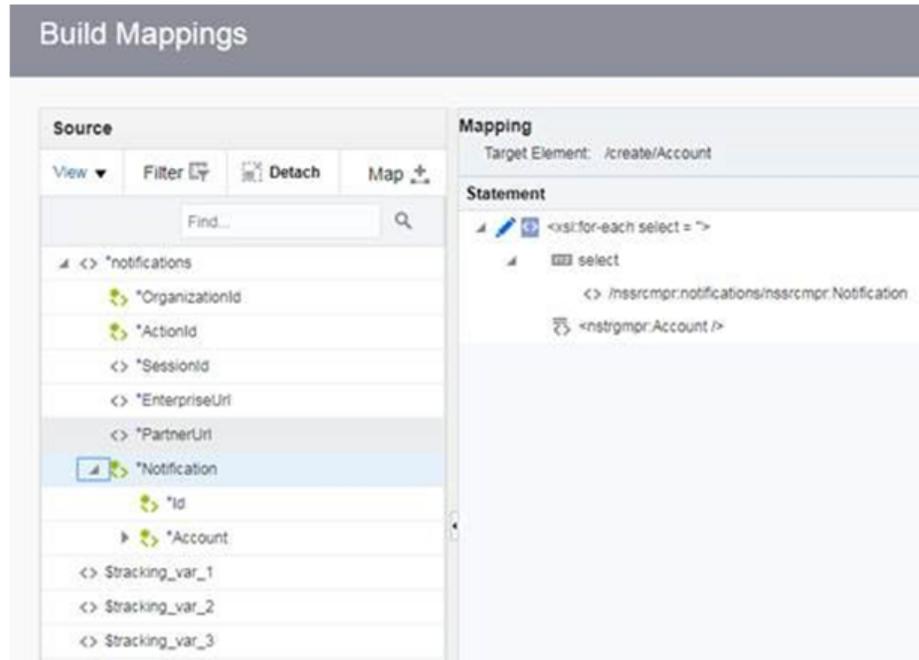


Send Multiple Records in a Single Message

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.

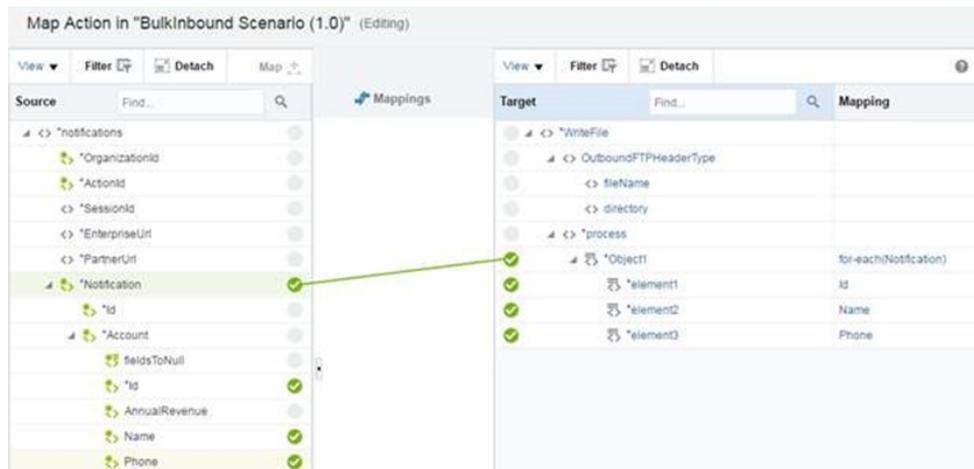
In the following scenario, the integration receives a single message containing multiple records. To support this scenario, you must configure the mapper when the request record shows a single occurrence, but contains multiple records. For this example, the steps to map the **Notification** element when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

Source	Find...	Q	Mappings	Target	Find...	Q	Mapping
<ul style="list-style-type: none"> <> *process Request InputString1 InputString2 				<ul style="list-style-type: none"> *create Account Name Phone 			<ul style="list-style-type: none"> for-each(Request) inputString1 inputString2

- Click **Validate**, then click **Close**.
- Configure response mapping.
 - Click the **Response** mapper icon for editing.
 - 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.
- Click **Repeat Element** to map both errors to the same element. The mappings are as follows:

Source	Find...	Q	Mappings	Target	Find...	Q	Mapping
<ul style="list-style-type: none"> *createResponse SaveResult errors extendedErrorDetails extendedErrorCode *message *statusCode DuplicateError extends errors extendedErrorDetails extendedErrorCode *message *statusCode *duplicateResult *allowSave *duplicateRule 				<ul style="list-style-type: none"> *processResponse Generic outputString1 outputString2 SuccessResponseError (1 of 2) extendedErrorDetails extendedErrorCode *message *statusCode SuccessResponseError (2 of 2) extendedErrorDetails extendedErrorCode *message *statusCode duplicateResult 			<ul style="list-style-type: none"> for-each(SaveResult) id success for-each(errors) for-each(extendedErrorDetails) extendedErrorCode message statusCode for-each(errors) for-each(extendedErrorDetails) extendedErrorCode message statusCode

- Click **Validate**, then click **Close**.
- Add a business tracking variable.
 - Click **Tracking**.
 - Map the business identifier as a tracking variable.
 - When complete, save and activate the integration.
 - 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.

Source	Find...	Mappings	Target	Find...	Mapping
<> *process			<> *create		
Request			Account		for-each(Request)
inputString1			Name		inputString1
inputString2			Phone		inputString2

- 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:

Source	Find...	Q	Mappings	Target	Find...	Q	Mapping
<> *message		✓		✓ <> *extendedErrorCode			extendedErrorCode
<> *statusCode		✓		✓ <> *message			message
▲ DuplicateError extends errors				✓ <> *statusCode			statusCode
▲ extendedErrorDetails				● ▶ duplicateResult			
<> *extendedErrorCode		✓		✓ ▲ SuccessResponseError (2 of 2)			for-each(errors)
fields				● ▲ extendedErrorDetails			
<> *message		✓		✓ <> *extendedErrorCode			extendedErrorCode
<> *statusCode		✓		✓ <> *message			message
▲ *duplicateResult				✓ <> *statusCode			statusCode
<> *allowSave		✓		● ▲ duplicateResult			
<> *duplicateRule		✓		✓ <> *allowSave			allowSave
<> *duplicateRuleEntityType				✓ <> *duplicateRule			duplicateRule
<> *errorMessage		✓		● <> *duplicateRuleEntityType			
▶ matchResults				✓ <> *errorMessage			errorMessage
xsi:type				● ▶ matchResults			

- 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.

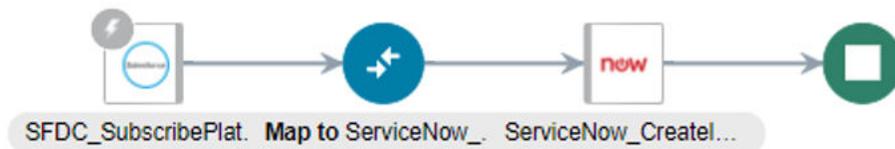
Synchronize Salesforce Platform Event Notifications (or Messages) With ServiceNow Incidents

The Salesforce Adapter supports receiving event notifications sent with platform events from the Salesforce application. You can subscribe to the platform events defined in

Salesforce.com and receive event notifications published into the channel as event messages.

This use case describes how the Salesforce Adapter is used to synchronize Salesforce platform event notifications with ServiceNow incidents. Configure a Salesforce trigger endpoint and select the platform event to receive the event notifications. Configure a ServiceNow invoke endpoint to create an incident using the create operation available in ServiceNow. This implementation pattern provides an overview of the steps.

1. Create an app-driven orchestrated integration.
2. Drag a Salesforce Adapter into the integration as a trigger connection.
3. Configure the Salesforce endpoint with the required platform event you defined in Salesforce.com.
 - a. On the Basic Info page, provide a name.
 - b. On the Action page, select **Platform Events**.
 - c. On the Platform Events page, select a platform event to receive event notifications from the Salesforce application.
 - d. Review your selections on the Summary page.
4. Drag a ServiceNow Adapter into the integration canvas.
5. Configure the ServiceNow endpoint with the required operation on the Operations page. For this example, the **Incident** operation is selected, which is triggered when event messages are published into the channel in the Salesforce application.
 - a. On the Basic Info page, provide an endpoint name.
 - b. On the Action page, select **Create, Update, or Delete Information**.
 - c. On the Operations page, select the **Incident** application, and then select the **Incident** operation.
 - d. Review your selections on the Summary page, and click **Done**.
6. In the mapper, perform the required mappings to create an incident in ServiceNow. The completed integration looks as follows.



7. Activate the integration.
 - a. At the end of the row for the integration, click the **Switch to Activate** icon.
 - b. If needed, check the **Enable Tracing** check box. The **Include Payload** check box is displayed.
 - c. If needed, check the **Include Payload** check box.
 - d. Click **Activate**. The integration is deployed.

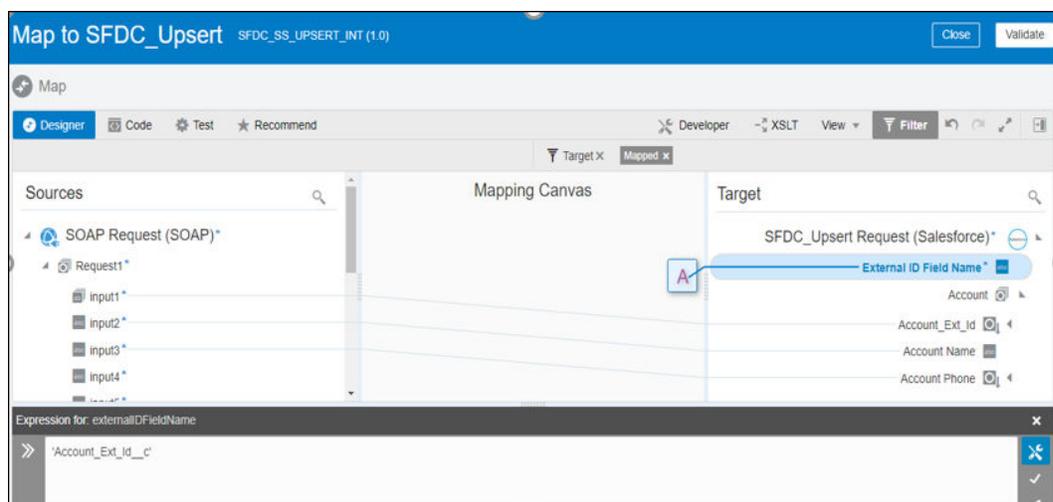
The integration is invoked when event messages are published into the channel in Salesforce.com and the message data from Salesforce.com is used to create an incident in ServiceNow.

Insert or Update Records Using the Upsert Operation

This use case provides an overview of how to use the upsert operation in an integration.

You can use the upsert operation to insert a new record or update an existing record in the Salesforce application. To determine if the record is already present, the upsert operation uses a unique key for matching the record. The unique key is a custom field declared as **External Id field** in the Salesforce application.

- If the key does not match, a new object record is created.
 - If the key matches once, the existing record of the object is updated automatically.
 - If the key matches more than once, an error is generated, and the object record is not inserted or updated.
1. Create SOAP Adapter and Salesforce Adapter connections.
 2. Create an app-driven orchestrated integration.
 3. Drag a SOAP Adapter connection into the integration as a trigger connection.
 4. Configure the SOAP endpoints.
 5. Drag a Salesforce Adapter into the integration canvas.
 6. Configure the Salesforce Adapter as follows.
 - a. On the Basic Info page, provide an endpoint name.
 - b. On the Action page, select **Create, Update or Delete Information**.
 - c. On the Operations page, select the **Upsert** operation, and then select the business object (for this example, **Account**).
 - d. Review your selections on the Summary page, and click **Done**.
 7. In the mapper, map the source **SOAP Request (SOAP)** input element nodes to the target **SFDC_Upsert Request (Salesforce)** input payload. For this example, map **input1** to **Account_Ext_Id** and **input2** to **Account Name**.

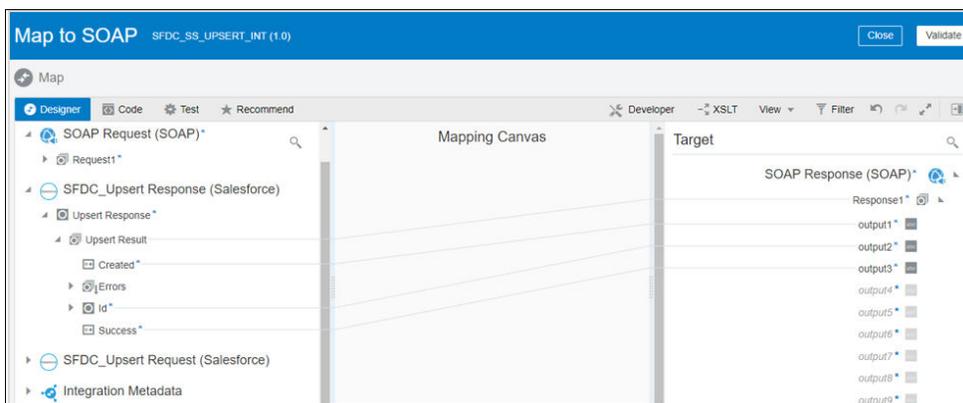


- Click **External ID Field Name**, and then click **Create target node**. The Expression Builder is displayed.
- Add the same field API name that is set as **External Id field** in Salesforce. For this example, the following value is added to the Expression Builder:

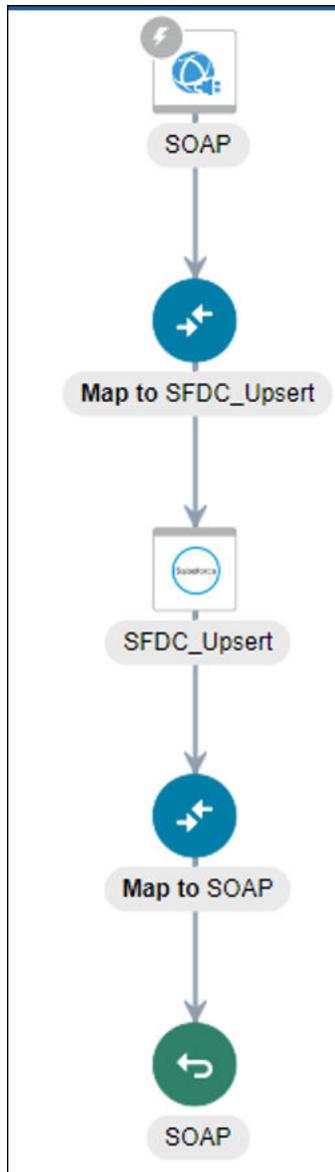
```
Account_Ext_id_c
```

Ensure that you use the same field name that you created/specified as an external ID field in the Salesforce application.

- In the mapper, map the **SFDC_Upsert Response (Salesforce)** element nodes to the **SOAP Response (SOAP)** element nodes:
 - **Created** to **output1** (returns **true** if a new account is created and **false** if an account is updated)
 - **Id** to **output2** (returns an account ID that is created or updated)
 - **Success** to **output3** (returns **true** if the operation is performed successfully)



- Click **Validate**, and then click **Close**. The completed integration looks as follows.



- 12. Specify the tracking variable.
- 13. Activate the integration.

5

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](#)
- [Troubleshoot the Platform Event Service in the Salesforce Adapter](#)
- [Error When the Upsert Operation is Performed Without Mapping the External ID Field](#)

Additional integration troubleshooting information is provided. See [Troubleshoot Oracle Integration in *Using Integrations in Oracle Integration Generation 2*](#) and the [Oracle Integration Troubleshooting page](#) on the Oracle Help Center.

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.

Troubleshoot the Platform Event Service in the Salesforce Adapter

The following issues can occur when you configure the Salesforce Adapter endpoint with the Platform Event service. Review the following table to identify solutions for these issues.

Issue	Solution
Why am I unable to see the notifications from the Salesforce application on the Tracking page after changing the platform event channel name?	An integration is bound to a specific channel. If you change/delete a platform event channel, all the activations/subscriptions become null-and-void and new events are not received and processed. In this case, you must set up the integration with the changed/new channel name.
Why am I unable to see the event notifications from the Salesforce application on the Tracking page?	<p>It may be possible that the maximum number of delivered event notifications within a 24-hour period exceeded the allocation limit for your Salesforce edition. See Salesforce Adapter Restrictions.</p> <p>You can make an API call to check the limit usage values of the delivered event notifications within a 24-hour period using the URL in the following format:</p> <pre>https://ap6.salesforce.com/services/data/api_version/limits</pre> <p>If the above API call confirms that the maximum number of delivered event notifications exceeded the allocation limit, no new event notification is received until the limits are restored. As soon as the limits are restored, the event notifications are received without any intervention. This information related to limit exceeding is logged in the diagnostic logs.</p>

Error When the Upsert Operation is Performed Without Mapping the External ID Field

When a Salesforce invoke endpoint is configured with the upsert operation on the Operations page of the Adapter Endpoint Configuration Wizard and an external ID field is not created/ specified in the Salesforce application or mapped in Oracle Integration, you get the following error when attempting to insert or update the record in the Salesforce application.

```
[oracle.soa.tracking.SCAEntityId: 143631] [composite_name:
SALESFORCE_ITEM_TEST!01.00.0000] [FlowId:
00000VzS50xBT8KpqS_AiY1^B1IO0006hu]
CloudAdapterInteraction#execute failed due to CASDK-0033: Received a SOAP
fault while invoking endpoint target:
https://tpbi--qa.sandbox.my.salesforce.com/services/Soap/c/
57.0/00D6u0000004YFL.
```

This indicates a processing exception on the service endpoint side. Please check service side logs to further diagnose the problem.

Solution: If you want to use the upsert operation to insert or update the records in the Salesforce application, create/specify an external ID field in the Salesforce application. The external ID field must also be mapped in the mapper. External ID is a mandatory parameter for the upsert operation.

A use case is provided for the upsert operation. See [Insert or Update Records Using the Upsert Operation](#) and [Use an 'External ID' to set the values for audit fields](#) for creation of external ID fields.