Oracle® Cloud

Using Integrations in Oracle Integration Generation 2

F47134-58
July 2024

Oracle Cloud Using Integrations in Oracle Integration Generation 2,

F47134-58

Primary Author: Oracle Corporation

Contents

5 Create Scheduled Integrations

10 Integrate with Processes and Services

11 Manage Integrations

12 Monitor Integrations During Runtime

13 Troubleshoot Oracle Integration

A Oracle Integration Use Cases

Preface

This guide describes how to use Oracle Integration to integrate your applications.

> **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

This guide is intended for users who want to create, activate, and monitor application integrations.

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 the Oracle Mapper*
- Adapter documentation in the Oracle Cloud Library on the Oracle Help Center

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Get Started with Integrations

Review information about integration concepts and components to help you get started creating your own integrations.

Topics:

Run the Sample Integrations

When you provision a new instance of Oracle Integration, the following sample integrations are automatically included. These samples help you get you up and running quickly and show you how easy it is to activate, invoke, and monitor an integration between endpoints.

> **Note:**
>
> If you provision a new instance of Oracle Integration Generation 2, these samples are *not* automatically included. Samples are provided at the Oracle Cloud Marketplace. See the Oracle Cloud Marketplace.

See the following sections to run and monitor the samples. It is recommended that you run these samples in the following order because several samples build upon one another.

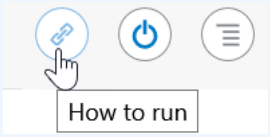
> **Note:**
>
> When you invoke an integration endpoint directly from the Chrome browser, it throws an `HTTP ERROR 401` error. The Chrome browser throws this error when an API supports both the Basic Authentication and the OAuth Authentication mechanisms. Use a different browser to run your integration.

Sample	Complexity	This Sample Demonstrates How to ...	See ...
Echo	None	Use simple logging with a REST Adapter as a trigger in a request-response orchestrated integration.	Run the Echo Sample
Hello World	None	Use simple log and email notification actions with a REST Adapter in an orchestrated integration.	Run the Hello World Sample
Hello World Invoke	Minimal	Call and invoke a simple REST web service in an orchestrated integration. Simple logging is also used.	Run the Hello World Invoke Sample
Hello World Data Map Invoke	Minimal	Call and invoke a simple REST web service using the Basic Routing integration style.	Run the Hello World Map Data Invoke Sample
File Transfer	Medium	Read an opaque file from a "`/`" directory and write the file to an "`/upload`" directory in a simple scheduled orchestration. After activating the integration, you go to the **Actions** menu and select **Submit now** or **Add Schedule** to run the scheduled integration.	Run the File Transfer Sample
Incident Details from Service Cloud	Medium	Get incident details from the Oracle Service Cloud for an incident ID and send the incident details to the caller as a response.	Run the Incident Details from Service Cloud Sample
Multiple Verbs and Resources Invoke	Medium	Create and invoke a REST web service with multiple HTTP verbs and resources.	Run the Multiple Verbs and Resources Invoke Sample
ConcatMessages	Low	Create and invoke a JavaScript callout action using a REST Adapter in an orchestrated integration.	Run the ConcatMessages Sample
Get Opportunity Details	Medium	Get opportunity details from the Oracle Engagement Cloud for an opportunity ID and send the opportunity details to the caller as a response.	Run the Get Opportunity Details Sample

Note:

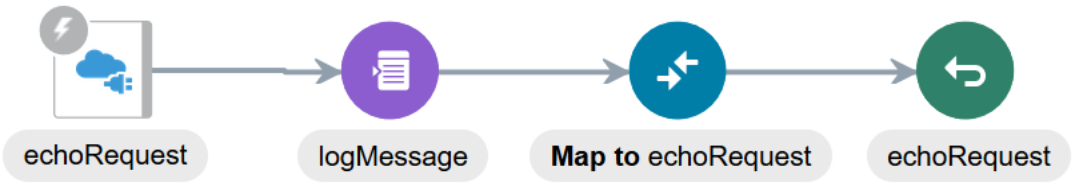
The documentation for running these samples is also available when you click

for an activated integration.



Run the Echo Sample

This sample demonstrates how to use simple logging with a REST Adapter as a trigger in a request-response orchestrated integration. The REST Adapter is triggered when you specify a URL. A browser response is sent to you. A logging message is created and logged to the activity stream for viewing. You also track the integration and monitor message status.



Complexity

None.

Prerequisites

None.

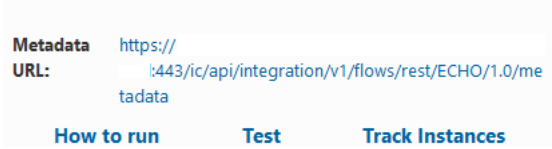
How To Activate

1. In the left navigation pane, click **Home** > **Integrations** > **Integrations**.
2. In the row for the Echo sample, click the
   
   icon, then click **Activate** when prompted.
3. Wait for the icon to turn green and the word **Active** to appear in the **Status** column, indicating that the integration is activated.

How To Run

1. Click the
   

icon to show a message with details about running, testing, and tracking the integration.



2. Enter the following URL in a browser. You can also access the URL from the

icon for this integration.

```
https://hostname:port/ic/api/integration/v1/flows/rest/ECHO/1.0/{message}
```

For example:

```
https://my_pod.us.company.com:port/ic/api/integration/v1/flows/rest/
ECHO/1.0/Invoking my first integration.
```

> **Note:**
>
> You can also run the integration by clicking **Test** to invoke the Test Integration page. See Test REST Adapter Trigger Connection-Based Integrations.

What Results Do You See

- You receive the following response in your browser:

```
{
  "Message" : "Invoking my first integration.",
  "Welcome" : "\"Welcome to OIC!!!  Echo was successful.\""
}
```

How To Monitor

1. In the left navigation pane, click **Home** > **Monitoring** > **Integrations** > **Dashboards**.
2. By default, the Dashboard page displays overall system status, including the percentage of successful messages, total number of messages, total number of successful messages, and total number of failed messages. Details about currently used connections, currently activated integrations, and scheduled integrations are also provided. You can also view the activity stream and download diagnostic logs and incident reports.
3. Select **Activity Stream** from the **View** menu to view details about the invocation.
4. In the left navigation pane, click **Tracking** and note that the **Echo** integration instance is listed as completed.
5. In the left navigation pane, click **Integrations** and note the status of the message processed.

How To View

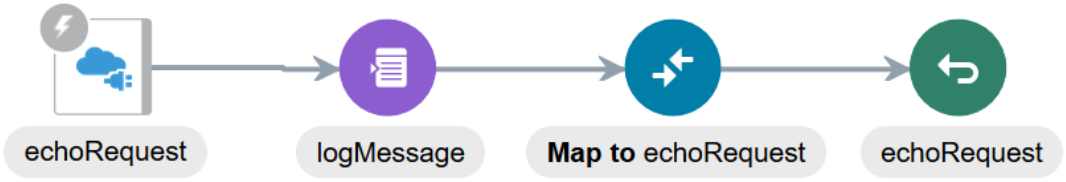
1. In the left navigation pane, click **Home** > **Integrations** > **Integrations**.

2. Click the **Echo** integration.

   A read-only version of the integration is displayed for viewing. Because the integration is active, it cannot be edited.

3. View the flow of the integration.

   - A REST Adapter is configured as a trigger (inbound) connection in the integration. The REST Adapter is configured with a resource endpoint of `/{message}` and a GET operation. The REST Adapter is triggered when you specify the URL in How to Run.
   - A logging message is created and logged to the activity stream. A browser response is sent to you.

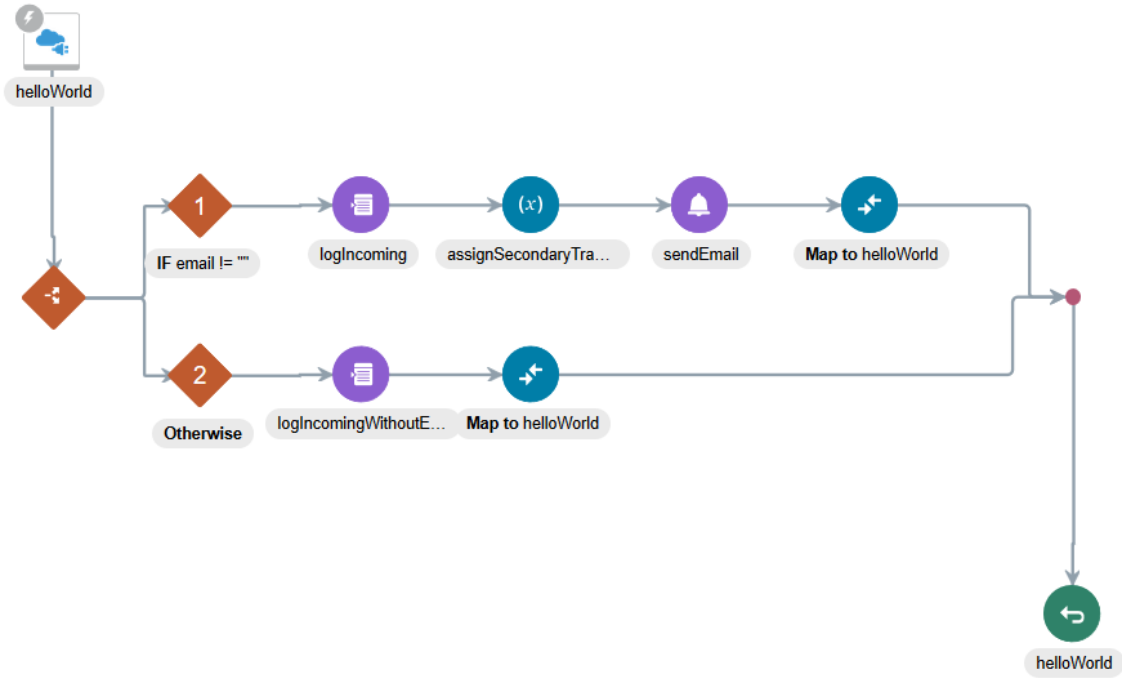


Learn More About The Features in this Sample

- Creating Orchestrated Integrations
- REST Adapter Capabilities
- Logging Messages with a Logger Action
- Getting Started with the Mapper

Run the Hello World Sample

This sample demonstrates how to use simple log and email notification actions with a REST Adapter in an orchestrated integration. The REST Adapter is triggered when you specify a URL. A switch activity with two rules is defined. If you provide an email address and name when triggering the REST Adapter, a browser response and email response are sent to you. If you provide only a name when triggering the REST Adapter, only a browser response is sent to you. With either switch rule, a logging message is created and logged to the activity stream for viewing. You also track the integration and monitor message status.



Complexity

None.

Prerequisites

None.

How To Activate

1. In the left navigation pane, click **Home** > **Integrations** > **Integrations**.
2. In the row for the Hello World sample, click the
   
   icon, then click **Activate** when prompted.
3. Wait for the icon to turn green and the word **Active** to appear in the **Status** column, indicating that the integration is activated.

How To Run

1. Click the
   
   icon to show a message with details about running, testing, and tracking the integration.
2. Enter one of the following URLs in a browser. You can also access these URLs from the
   
   icon for this integration.

   ```
   http://hostname:port/ic/api/integration/v1/flows/rest/HELLO_WORLD/1.0/
   names/{name}
   http://hostname:port/ic/api/integration/v1/flows/rest/HELLO_WORLD/1.0/
   names/{name}?email={email.address}
   ```

For example:

```
https://my_pod.us.company.com:port/ic/api/integration/v1/flows/rest/
HELLO_WORLD/1.0/names/Mark

https://my_pod.us.company.com:port/ic/api/integration/v1/flows/rest/
HELLO_WORLD/1.0/names/Mark?email=mark.smith@mycompany.com
```

> **Note:**
>
> You can also run the integration by clicking **Test** to invoke the Test Integration page. See Test REST Adapter Trigger Connection-Based Integrations.

What Results Do You See

- If you specified only your name, you receive the following response in your browser:

```
{
  "Hello" : "Mark",
  "Message" : "\"Welcome to OIC!!!\"",
  "Email" : "\"Email address was not provided.\""
}
```

- If you specified your name and email address, you receive the following response in your browser:

```
{
  "Hello" : "Mark",
  "Message" : "\"Welcome to OIC!  Check your email.\"",
  "Email" : "mark.smith@mycompany.com"
}
```

and you receive a Hello email with the following contents:

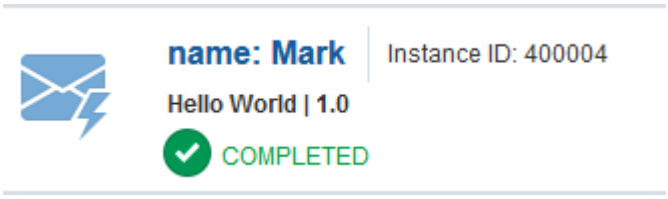
```
Hello Mark,

Welcome to Oracle Integration!
```

How To Monitor

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

   By default, the Dashboard page displays overall system status, including the percentage of successful messages, total number of messages, total number of successful messages, and total number of failed messages. Details about currently used connections, currently activated integrations, and scheduled integrations are also provided. You can also view the activity stream and download diagnostic logs and incident reports.
2. Select **Activity Stream** from the **View** menu to view details about the invocation.
3. In the left navigation pane, click **Tracking** and note that the **Hello World** integration instance is listed as completed.

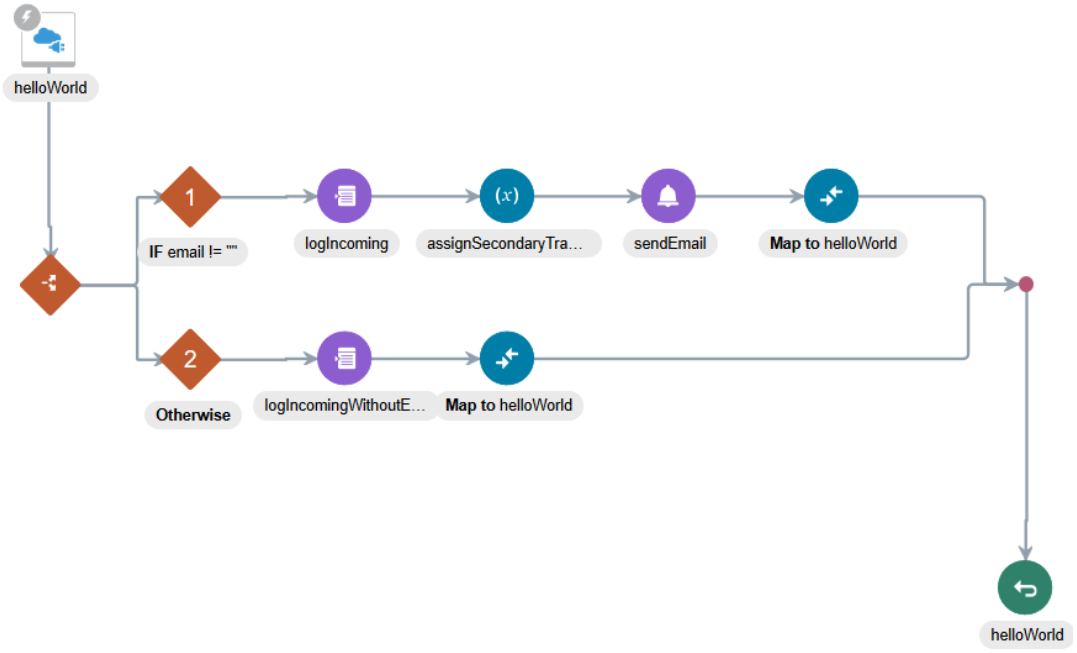


4. In the navigation pane, click **Integrations** and note that the message was successfully received and processed without any errors.

How To View

1. In the left navigation pane, click **Home** > **Integrations** > **Integrations**.
2. Click the **Hello World** integration.

   A read-only version of the integration is displayed for viewing. Because the integration is active, it cannot be edited.
3. View the flow of the integration:
   - A REST Adapter is configured as a trigger (inbound) connection in the integration. The REST Adapter is configured with a resource endpoint of `/names{name}` and a GET operation. The REST Adapter is triggered when you specify the URL in How to Run.
   - A switch activity with two rules is defined:
     - If an email address and name are provided in the invocation URL (upper rule), a logging message is created and logged to the activity stream and a notification action is configured with parameters for your name and email address. The message body of the email is also defined in the notification action. A browser response and email response are sent to you.
     - If only a name is provided in the invocation (lower rule), a logging message is created and logged to the activity stream. A browser response is sent to you.

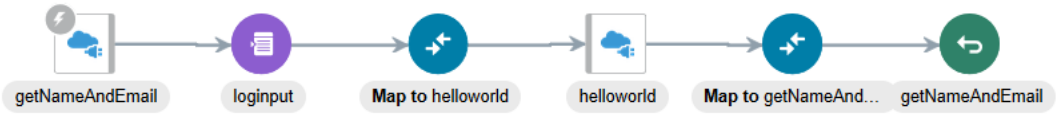


Learn More About The Features in this Sample

- Creating Orchestrated Integrations
- Defining Switch Branches
- REST Adapter Capabilities
- Logging Messages with a Logger Action
- Getting Started with the Mapper

Run the Hello World Invoke Sample

This sample demonstrates how to call and invoke a simple REST web service in an orchestrated integration. The REST Adapter is triggered when you specify a URL. The **Hello World** REST Adapter you previously used in the **Hello World** sample is invoked in the integration. The name and email address response are sent to you in JSON format. A logging message is created and logged to the activity stream for viewing. You also track the integration and monitor message status.



Complexity

Minimal.

Prerequisites

The Hello World sample integration must be activated before configuring the connection in How to Configure.

How to Configure

Before you can activate and run this sample, you must configure the connection and security properties of the **ORACLE REST Sample Hello World Invoke** REST Adapter used in this sample.

1. In the left navigation pane, click **Home** > **Integrations** > **Connections**.
2. Click **ORACLE REST Sample Hello World Invoke**.
3. Go to the **Connection Properties** section to specify information to connect to the application/endpoint and process requests.
   a. For **Connection Type**, select **REST API Base URL**.
   b. For **Connection URL**, enter the following:

      ```
      https://hostname:port
      ```

4. Go to the **Security** section.
   a. Enter the username and password that you used to log in to Oracle Integration, then click **Save**.
5. Click **Test** to test your configuration. A message is displayed that describes the results of the test. If successful, you are ready to activate the integration.

   ```
   Connection ORACLE REST Sample Hello World Invoke was tested successfully.
   ```

6. Click **Save**, then click
   
   .

How To Activate

1. In the row for the Hello World Invoke sample, click the
   
   icon, then click **Activate** when prompted.
2. Wait for the icon to turn green and the word **Active** to appear in the **Status** column, indicating that the integration is activated.

How To Run

1. Click the
   
   icon to show a message with details about running, testing, and tracking the integration.
2. Enter one of the following URLs in a browser. You can also access these URLs from the
   
   icon for this integration.

   ```
   https://host:port/ic/api/integration/v1/flows/rest/HELLO_WORLD_INVOKE/1.0/
   info?name={Name}
   ```

   ```
   https://host:port/ic/api/integration/v1/flows/rest/HELLO_WORLD_INVOKE/1.0/
   info?name={Name}&email={Email Address}
   ```

For example:

```
https://my_pod.us.company.com:port/ic/api/integration/v1/flows/rest/
HELLO_WORLD_INVOKE/1.0/info?name=Mark
https://my_pod.us.company.com:port/ic/api/integration/v1/flows/rest/
HELLO_WORLD_INVOKE/1.0/info?name=Mark&email=mark.smith@mycompany.com
```

> **Note:**
>
> You can also run the integration by clicking **Test** to invoke the Test Integration page. See Test REST Adapter Trigger Connection-Based Integrations.

What Results Do You See

If you specified a name and email, you receive the following response in your browser:

```
{
  "Hello" : "mark",
  "Message" : "\"Welcome to OIC!  Check your email.\"You have successfully
called a REST service!",
  "Email" : "mark.smith@mycompany.com"
}
```

and you receive a Hello email with the following contents:

```
Hello Mark,

Welcome to OIC!
```

How To Monitor

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

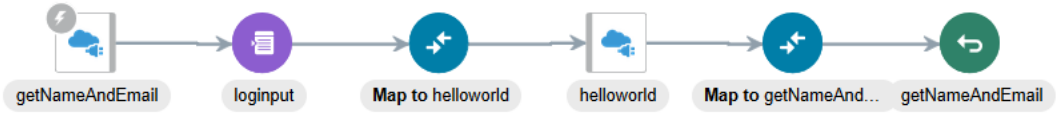
   By default, the Dashboard page displays overall system status, including the percentage of successful messages, total number of messages, total number of successful messages, and total number of failed messages. Details about currently used connections, currently activated integrations, and scheduled integrations are also provided. You can also view the activity stream and download diagnostic logs and incident reports.
2. Select **Activity Stream** from the **View** menu to view details about the invocation.
3. In the left navigation pane, click **Tracking** and note that **Hello World Invoke** is listed as completed. **Hello World**, which was invoked by **Hello World Invoke**, has also completed.
4. In the navigation pane, click **Integrations** and note that the messages were successfully received and processed without any errors.

How To View

1. In the left navigation pane, click **Home** > **Integrations** > **Integrations**.
2. Click the **Hello World Invoke** integration.

   A read-only version of the integration is displayed for viewing. Because the integration is active, it cannot be edited.
3. View the flow of the integration:

- A REST Adapter is configured as a trigger (inbound) connection in the integration. The REST Adapter is configured with a resource endpoint of `/info` and a GET operation, and retrieves a name and email address. This REST Adapter is triggered when you specify the URL in How to Run.
- A logging message is created and logged to the activity stream.
- The **Hello World** REST Adapter you previously used in the **Hello World** sample is invoked in the integration. The REST Adapter is configured with a business object of `/name/{name}`, a GET operation, and a request query parameter of `email`. The name and email address response are sent to you in JSON format.

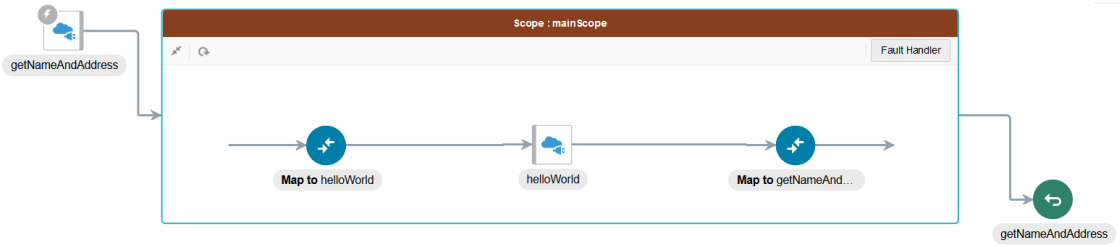


Learn More About The Features in this Sample

- Creating Orchestrated Integrations
- REST Adapter Capabilities
- Logging Messages with a Logger Action
- Getting Started with the Mapper

Run the Hello World Map Data Invoke Sample

This sample demonstrates how to call and invoke a simple REST web service using the Basic Routing integration style. The REST Adapter is triggered when you specify a URL. The **Hello World Invoke** REST Adapter is invoked in the integration. The name and email address response are sent to you in JSON format. A logging message is created and logged to the activity stream for viewing. You also track the integration and monitor message status.



Complexity

Minimal.

Prerequisites

The Hello World sample integration must be activated before configuring the connection in How to Configure.

How to Configure

Before you can activate and run this sample, you must configure the connection and security properties of the **ORACLE REST Sample Hello World Invoke** REST Adapter used in this sample.

1. In the left navigation pane, click **Home** > **Integrations** > **Connections**.
2. Click **ORACLE REST Sample Hello World Invoke**.
3. Go to the **Connection Properties** section to specify information to connect to the application/endpoint and process requests.
   a. For **Connection Type**, select **REST API Base URL**.
   b. For **Connection URL**, enter the following:

      ```
      https://hostname:port
      ```

4. Go to the **Security** section.
   a. Enter the username and password that you used to log in to Oracle Integration, then click **OK**.
5. Click **Test** to test your configuration. A message is displayed that describes the results of the test. If successful, you are ready to activate the integration.

   ```
   Connection ORACLE REST Sample Hello World Invoke was tested successfully.
   ```

6. Click **Save**, then click
   
   .

How To Activate

1. In the left navigation pane, click **Home** > **Integrations** > **Integrations**.
2. In the row for the Hello World Data Map Invoke sample, click the
   
   icon, then click **Activate** when prompted.
3. Wait for the icon to turn green and the word **Active** to appear in the **Status** column, indicating that the integration is activated.

How To Run

1. Click the
   
   icon to show a message with details about running, testing, and tracking the integration.
2. Enter one of the following URLs in a browser.

   ```
   https://hostname:port/ic/api/integration/v1/flows/rest/
   HELLO_WORLD_MAP_DATA_INVOKE/1.0/info?name={Name}
   https://hostname:port/ic/api/integration/v1/flows/rest/
   HELLO_WORLD_MAP_DATA_INVOKE/1.0/info?name={Name}&email={Email Address}
   ```

For example:

```
https://my_pod.us.company.com:port/ic/api/integration/v1/flows/rest/
HELLO_WORLD_MAP_DATA_INVOKE/1.0/info?name=mark
https://my_pod.us.company.com:port/ic/api/integration/v1/flows/rest/
HELLO_WORLD_MAP_DATA_INVOKE/1.0/info?
name=mark&email=mark.smith@mycompany.com
```

> **Note:**
>
> You can also run the integration by clicking **Test** to invoke the Test Integration page. See Test REST Adapter Trigger Connection-Based Integrations.

What Results Do You See

- If you specified only your name, you receive the following response in your browser:

```
{
  "Hello" : "mark",
  "Message" : "\"Welcome to OIC!!!\"You have successfully called a REST
service!",
  "Email" : "\"Email address was not provided.\""
}
```

- If you specified your name and email address, you receive the following response in your browser:

```
{
  "Hello" : "mark",
  "Message" : "\"Welcome to OIC!  Check your email.\"You have successfully
called a REST service!",
  "Email" : "mark.smith@mycompany.com"
}
```

and you receive a Hello email with the following contents:

```
Hello mark,
Welcome to OIC !
```

How To Monitor

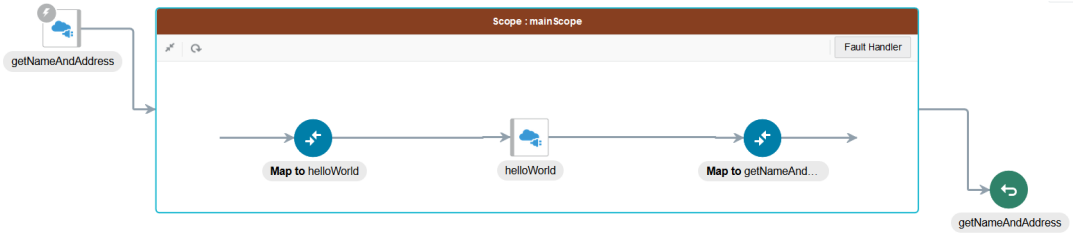
1. In the left navigation pane, click **Home** > **Monitoring** > **Integrations** > **Dashboards**.
2. By default, the Dashboard page displays overall system status, including the percentage of successful messages, total number of messages, total number of successful messages, and total number of failed messages. Details about currently used connections, currently activated integrations, and scheduled integrations are also provided. You can also view the activity stream and download diagnostic logs and incident reports.
3. Select **Activity Stream** from the **View** menu to view details about the invocation.
4. In the left navigation pane, click **Tracking** and note that the **Hello World Map Data Invoke** instance is listed as completed. **Hello World**, which was invoked by **Hello World Map Data Invoke**, has also completed.

5. In the left navigation pane, click **Integrations** and note that the messages were successfully received and processed without any errors.

How To View

1. In the left navigation pane, click **Home** > **Integrations** > **Integrations**.
2. Click the **Hello World Data Map Invoke** integration.

   A read-only version of the integration is displayed for viewing. Because the integration is active, it cannot be edited.
3. View the flow of the integration:
   - A REST Adapter is configured as a trigger (inbound) connection in the integration. The REST Adapter is configured with a resource endpoint of `/info` and a GET operation, and retrieves a name and email address. The REST Adapter is triggered when you specify the URL in How to Run.
   - A logging message is created and logged to the activity stream.
   - The **Hello World Invoke** REST Adapter is invoked in the integration. The REST Adapter is configured with a business object of `/name/{name}`, a GET operation, and a request query parameter of `email`. The name and email address response are sent to you in JSON format.

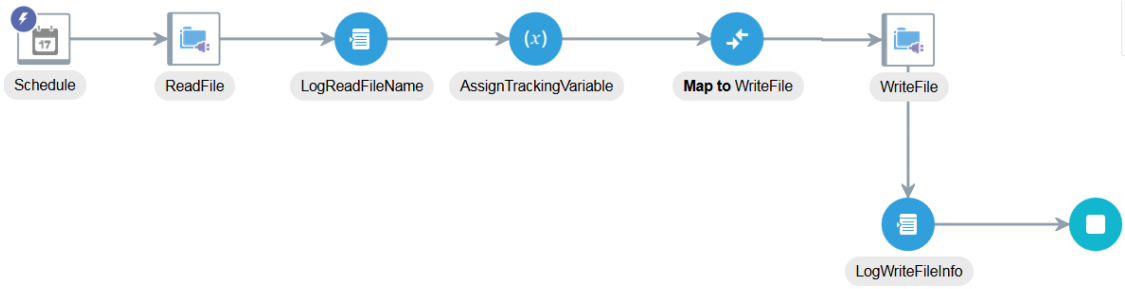


Learn More About The Features in this Sample

- Understand Integration Creation and Best Practices
- REST Adapter Capabilities
- Getting Started with the Mapper

Run the File Transfer Sample

This sample demonstrates how to read an opaque file from a "`/`" directory and write the file to an "`/upload`" directory in a scheduled orchestrated integration. An FTP Adapter reads the file from the `/` directory and another FTP Adapter writes the file to the `/upload` directory. An assign action is configured to assign variables for the file name and file size. A logging message is created to indicate that the file name has been read. The message is logged to the activity stream for viewing. You also track the integration and monitor message status.



Complexity

Medium.

Prerequisites

None.

How To Activate

1. In the left navigation pane, click **Home** > **Integrations** > **Integrations**.
2. In the row for the File Transfer sample, click the
   
   icon, then click **Activate** when prompted.
3. Wait for the icon to turn green and the word **Active** to appear in the **Status** column, indicating that the integration is activated.

How to Configure

Before you can activate and run this sample, you must configure the connection and security properties of the **Sample FTP connection** FTP Adapter used in this sample.

1. In the left navigation pane, click **Home** > **Integrations** > **Connections**.
2. Click **Sample FTP connection**.
3. Go to the **Connection Properties** section to specify information to connect to the application/endpoint and process requests.
   a. For **FTP Server Host Address**, enter `speedtest.tele2.net`.
   b. For **FTP Server Port**, enter `21`.
4. Go to the **Security** section.
   a. For **Security Policy**, select **FTP Server Access Policy**.
   b. For **Username**, enter `anonymous`.
   c. Enter the same password twice. You can use any password.
   d. Click **Save**.
5. Click **Test** to test your configuration. A message is displayed that describes the results of the test. If successful, you are ready to activate the integration.

   ```
   Connection Sample FTP connection was tested successfully.
   ```

6. Click **Save**, then click

.

How To Run

1. Click the

icon to show a message with details about running, testing, and tracking the integration.
2. Click **Submit Now**.
The Submit Now dialog is displayed.
3. Click **Submit Now**.
The following message is displayed at the top of the page.

```
 CONFIRMATION
Submit Now request to run integration File Transfer Sample
(version_number) was submitted successfully with request id number.
```

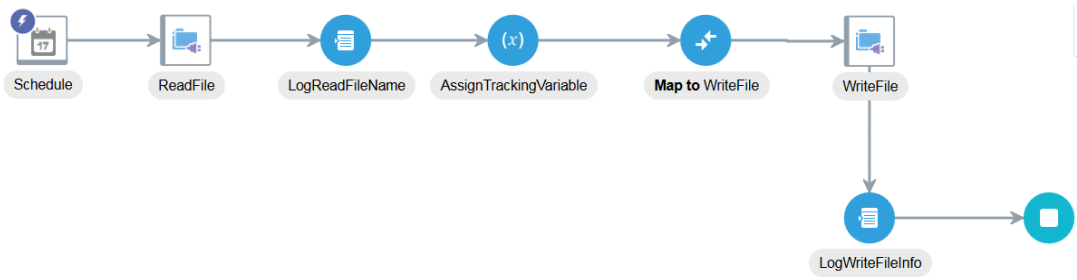
How To Monitor

1. Click the instance number in the message at the top to go to the Track Instances page. The **File Transfer sample** integration instance is listed as in progress. You can see the filename and file size, which enables you to look for particular files in case of problems.
2. Wait for the run to complete.
3. In the left navigation pane, click **Integrations** and note that the message was successfully received and processed without any errors. For this example, **Schedule is not defined** is displayed because this scheduled integration was invoked with the **Submit Now** option.

How To View

1. In the left navigation pane, click **Home** > **Integrations** > **Integrations**.
2. Click the **File Transfer** integration.

   A read-only version of the integration is displayed for viewing. Because the integration is active, it cannot be edited.
3. View the flow of the integration:
   - The **Schedule** icon indicates that this is a scheduled orchestrated integration. The schedule invokes this integration.
   - An FTP Adapter (**ReadFile**) is configured to read a file in binary mode from the `/` directory of the server you specified in the Connections page. No schema is defined for this file transfer, so it is treated as an attachment.
   - A logging message is created to indicate that the file name has been read. The message is logged to the activity stream.
   - An assign action is configured to assign variables for the file name and file size. This enables you to track issues based on filename and size.
   - An FTP Adapter (**WriteFile**) is configured to write a file to the `/upload` directory on the same server that matches the file name pattern of `1KB%yyMMddHHmmssSS%.zip`.
   - A logging message is created to indicate the file name has been uploaded to the `/upload` directory. The message is logged to the activity stream.

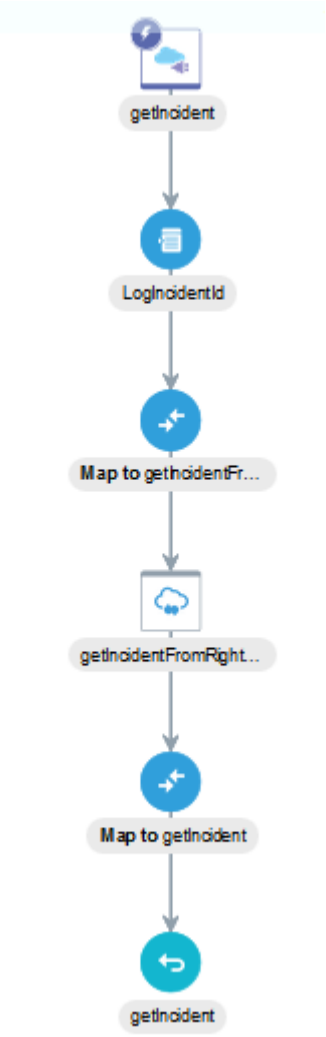


Learn More About The Features in this Sample

- Scheduling Integration Runs
- FTP Adapter Capabilities
- Creating Orchestrated Integrations
- Logging Messages with a Logger Action
- Assigning Values to Scalar Variables in an Assign Action
- Getting Started with the Mapper

Run the Incident Details from Service Cloud Sample

This sample demonstrates how to get incident details from the Oracle Service Cloud for an incident ID and send the incident details to the caller as a response. The REST Adapter is triggered when you specify a URL. The Oracle Service Cloud (RightNow) Adapter is invoked in the integration. The Oracle Service Cloud (RightNow) Adapter is configured to get incident details from the Oracle Service Cloud. The incident response is returned to you. A logging message is created and logged to the activity stream for viewing. You also track the integration and monitor message status.



Complexity

Medium.

Prerequisites

To run this sample, you must first subscribe to Oracle RightNow Cloud and configure an Oracle Service Cloud (RightNow) Adapter connection. After subscribing, you receive the WSDL, username, and password to specify for the Oracle Service Cloud (RightNow) Adapter on the Connections page. See Creating an Oracle Service Cloud (RightNow) Adapter Connection.

How To Activate

1. In the left navigation pane, click **Home** > **Integrations** > **Integrations**.
2. In the row for the Incident Details from Service Cloud sample, click the
   
   icon, then click **Activate** when prompted.
3. Wait for the icon to turn green and the word **Active** to appear in the **Status** column, indicating that the integration is activated.

How To Run

Enter the following URL in a browser with a valid incident ID value:

```
https://hostname:port/ic/api/integration/v1/flows/rest/
SAMPL_INCID_DETAI_FROM_SERVI_CLO/1.0/incident/12
```

For example:

```
https://my_pod.us.company.com:port/ic/api/integration/v1/flows/rest/
SAMPL_INCID_DETAI_FROM_SERVI_CLO/1.0/incident/12
```

What Results Do You See

You receive the following incident details response in your browser from the Oracle Service Cloud:

```
{
  "IncidentName" : "Survey Feedback",
  "LookupName" : "111206-000001",
  "CreatedTime" : "2011-12-06T22:35:11.000Z",
  "UpdatedTime" : "2011-12-06T22:35:11.000Z"
}
```

How To Monitor

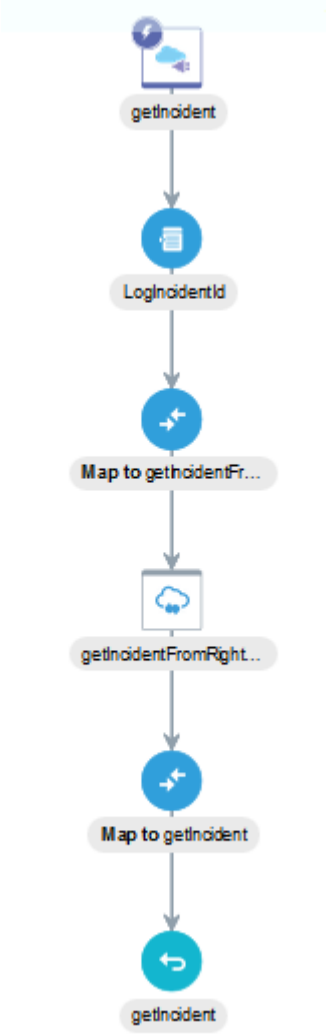
1. In the left navigation pane, click **Home** > **Monitoring** > **Integrations** > **Dashboards**.
   By default, the Dashboard page displays overall system status, including the percentage of successful messages, total number of messages, total number of successful messages, and total number of failed messages. Details about currently used connections, currently activated integrations, and scheduled integrations are also provided. You can also view the activity stream and download diagnostic logs and incident reports.
2. Select **Activity Stream** from the **View** menu to view details about the incident ID.
3. In the left navigation pane, click **Tracking** and note that the **Incident details from Service Cloud** integration instance is listed as completed.
4. In the left navigation pane, click **Integrations** and note that the message was successfully received and processed without any errors.

How To View

1. In the left navigation pane, click **Home** > **Integrations** > **Integrations**.
2. Click the **Incident details from Service Cloud** integration.

   A read-only version of the integration is displayed for viewing. Because the integration is active, it cannot be edited.
3. View the flow of the integration:
   - A REST Adapter is configured as a trigger (inbound) connection in the integration. The REST Adapter is configured with a resource endpoint of `/incident{id}` and a GET operation, and retrieves the incident ID. This REST Adapter is triggered when you specify the URL in How to Run.
   - A logging message is created and logged to the activity stream.

- The Oracle Service Cloud (RightNow) Adapter is invoked in the integration. The adapter is configured with an incident business object and a CRUD Get operation to get incident details from the Oracle Service Cloud. The incident response is returned to you.



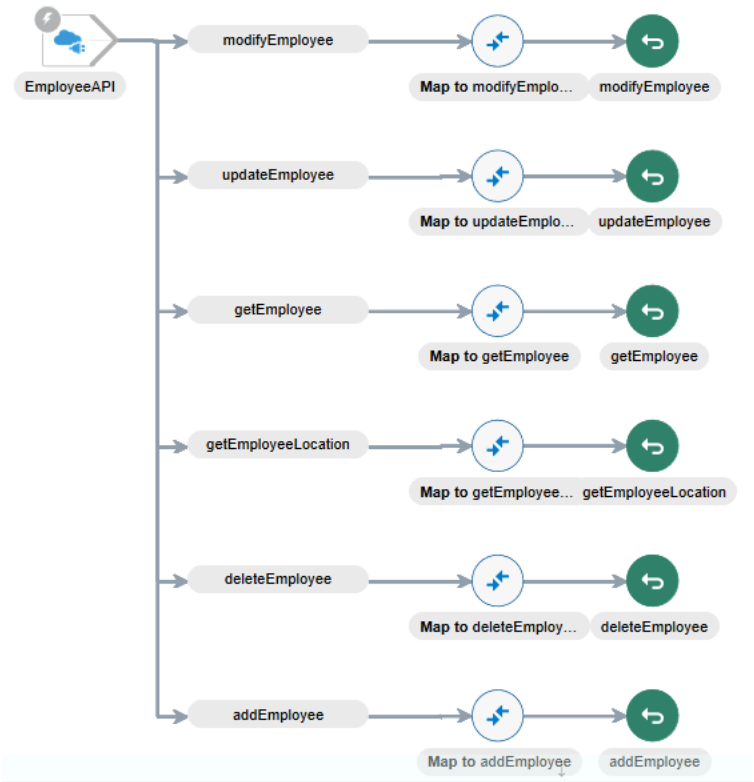
Learn More About The Features in this Sample

- Creating Orchestrated Integrations
- REST Adapter Capabilities
- Logging Messages with a Logger Action
- Getting Started with the Mapper
- Oracle Service Cloud (RightNow) Adapter Capabilities

Run the Multiple Verbs and Resources Invoke Sample

This sample demonstrates how to use multiple HTTP verbs and resources configured in the REST Adapter trigger connection of an integration. A single REST Adapter can be configured to have multiple HTTP verbs such as GET, PUT, POST, PATCH, and DELETE. The REST

Adapter can also be configured with multiple resources. For this sample, a single /emp resource and an /emp/loc subresource are used.



Complexity

Medium.

Prerequisites

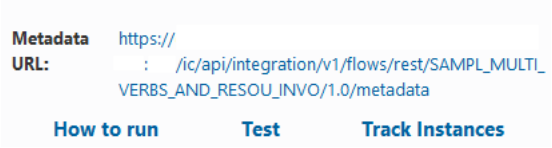
None.

How To Activate

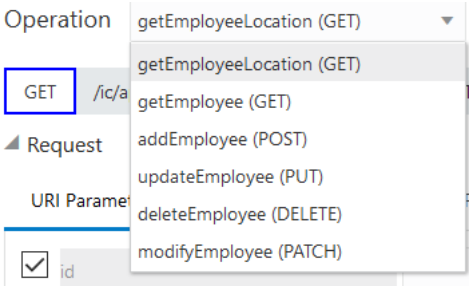
1. In the left navigation pane, click **Home** > **Integrations** > **Integrations**.
2. In the row for the **Sample Multiple Verbs and Resources Invoke** sample, click the
   
   icon, then click **Activate** when prompted.
3. Wait for the icon to turn green and the word **Active** to appear in the **Status** column, indicating that the integration is activated.

How To Run

1. Click the
   
   icon to display a message with details about running, testing, and tracking the integration.



2. Click **Test**.
3. On the Test Integration page, click the **Operation** dropdown list to view all configured options and HTTP verbs.
4. From the dropdown list, select the operation to perform.



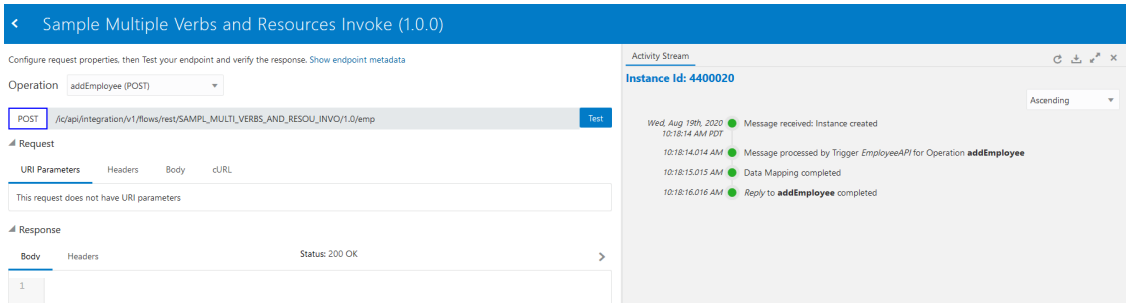
The section below the dropdown list shows all the options available for the selected operation such as **URI Parameters** and **Body**.

5. Click **Test** to run the selected operation.

What Results Do You See

The activity stream opens and shows the execution path of the request.

The **Response** section indicates that the invocation succeeded with a status of **200 OK**.



How To Monitor

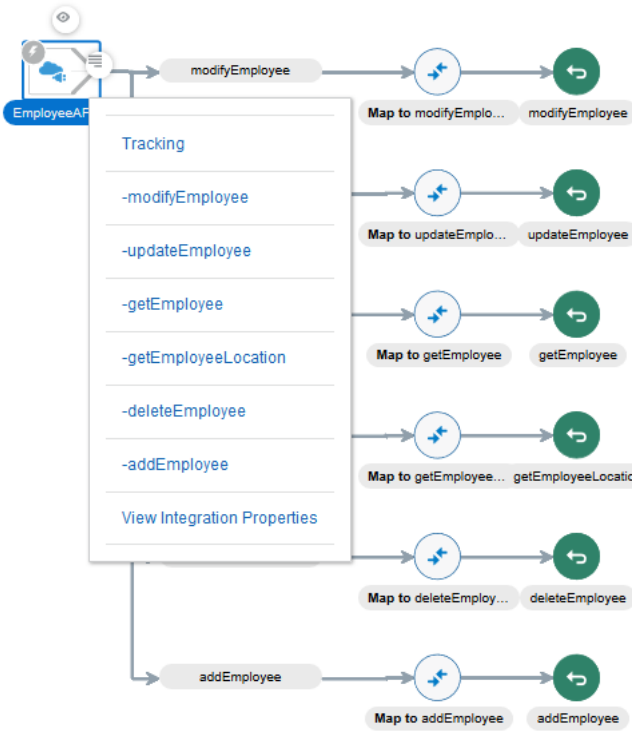
In addition to viewing the activity stream that appears on the Test Integration page, you can click the instance ID that appears above the stream to open the Tracking Details page. This page provides a graphical view of tracking instance details and the exact path (in the color green) followed by the request in the integration.

Click **Close** to navigate back to the Test Integration page. Another operation can be selected from the dropdown list and invoked by clicking **Test**.

How To View

1. In the left navigation pane, click **Home** > **Integrations** > **Integrations**.
2. Click the **Sample Multiple Verbs and Resources Invoke** integration.

   A read-only version of the integration is displayed for viewing. Because the integration is active, it cannot be edited.
3. View the flow of the integration.
   - A REST Adapter (**EmployeeAPI**) is configured as a trigger (inbound) connection in the integration. The REST Adapter is configured with a resource endpoint of /emp, a subresource of /emp/loc, and multiple verbs: GET, PUT, POST, PATCH, and DELETE operations. The configured operations are **modifyEmployee** (**PATCH**), **updateEmployee** (**PUT**), **getEmployee** (**GET**), **getEmployeeLocation** (**GET**), **deleteEmployee** (**DELETE**), and **addEmployee** (**POST**).
   - The REST Adapter is triggered when you specify the URL in How to Run.
   - Each operation is shown as a different path that branches out from the trigger connection.
   - Each branch has a map and a reply action configured.
   - The tracking configuration for each branch can be viewed by clicking the menu option on the REST Adapter trigger connection and selecting the appropriate operation from the dropdown list.

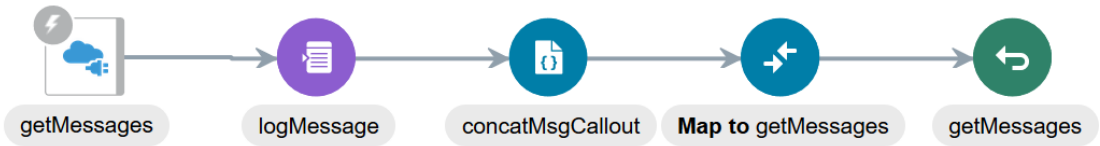


Learn More About The Features in this Sample

- Creating Orchestrated Integrations
- Receive Requests for Multiple Resources in a Single REST Adapter Trigger Connection
- Test REST Adapter Trigger Connection-Based Integrations
- Getting Started with the Mapper

Run the ConcatMessages Sample

This sample demonstrates how to use a simple callout action with a REST Adapter in an orchestrated integration. The REST Adapter is triggered when you specify a URL with two string query parameters. These parameters are then concatenated by a JavaScript callout action and a string message is returned as a response. A logging message is created and logged to the activity stream for viewing. You can also track the integration and monitor message status.



Complexity

Low.

Prerequisites

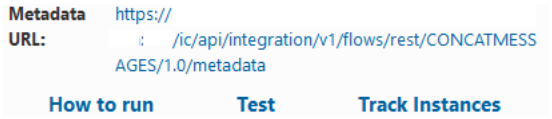
None.

How To Activate

1. In the left navigation pane, click **Home** > **Integrations** > **Integrations**.
2. In the row for the **ConcatMessages** sample, click the
   
   icon, then click **Activate** when prompted.
3. Wait for the icon to turn green and the word **Active** to appear in the **Status** column, indicating that the integration is activated.

How To Run

1. Click the
   
   icon to show a message with details about running, testing, and tracking the integration.



2. Enter the following URL in a browser. You can also access this URL from the **Metadata URL** field at the top of the message.

   ```
   https://hostname:port/ic/api/integration/v1/flows/rest/CONCATMESSAGES/1.0/
   welcome?message1=Welcome to OIC,&message2=Sample ConcatMessage Integration
   ```

   For example:

   ```
   https://mypod.us.company.com:port/ic/api/integration/v1/flows/rest/
   CONCATMESSAGES/1.0/welcome?message1=Welcome to OIC,&message2=Sample
   ConcatMessage Integration
   ```

What Results Do You See

1. If you have specified `message1` and `message2`, you receive the following response in your browser:

   ```
   Message : "Welcome to OIC, Sample ConcatMessage Integration"
   ```

   .

How To Monitor

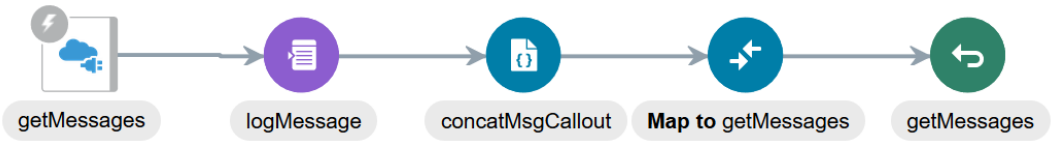
1. In the left navigation pane, click **Home** > **Monitoring** > **Integrations** > **Dashboards**.
   By default, the Dashboard page displays overall system status, including the percentage of successful messages, total number of messages, total number of successful messages, and total number of failed messages. Details about currently activated integrations and

scheduled integrations are provided. You can also view the activity stream and download diagnostic logs and incident reports.

2. Select **Activity Stream** from the **View** menu to view details about the invocation.
3. In the left navigation pane, click **Tracking** and note that the **ConcatMessages** integration instance is listed as completed.
4. In the left navigation pane, click **Integrations** and note the status of the messages processed.

How To View

1. In the left navigation pane, click **Home** > **Integrations** > **Integrations**.
2. Click the **ConcatMessages** integration.
3. View the flow of the integration.
   - A REST Adapter is configured as a trigger (inbound) connection in the integration. The REST Adapter is configured with a **/welcome** resource endpoint, a **GET** operation, and the **message1** and **message2** request query parameters. The REST Adapter is triggered when you specify the URL in How to Run.
   - A logging message is created and logged to the activity stream.
   - A callout action concatenates the input parameters **message1** and **message2** and a response is sent to you in JSON format.

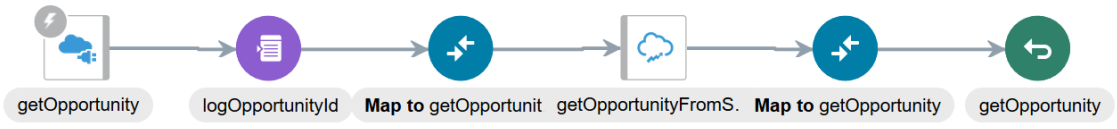


Learn More About The Features in this Sample

- Creating Orchestrated Integrations
- Add a JavaScript Action
- Getting Started with the Mapper

Run the Get Opportunity Details Sample

This sample demonstrates how to get opportunity details from Oracle Engagement Cloud for an opportunity ID and send the opportunity details to the caller as a response. The REST Adapter is triggered when you specify a URL. The Oracle CX Sales and B2B Service Adapter is invoked in the integration. The Oracle CX Sales and B2B Service Adapter is configured to get opportunity details from Oracle Engagement Cloud. The opportunity response is returned to you. A logging message is created and logged to the activity stream for viewing. You also track the integration and monitor message status.



Complexity

Medium.

Prerequisites

To run this sample, you must first subscribe to Oracle Engagement Cloud and configure an Oracle CX Sales and B2B Service Adapter connection. After subscribing, you receive the WSDL, username, and password to specify for the Oracle CX Sales and B2B Service Adapter on the Connections page.

How To Activate

1. In the left navigation pane, click **Home** > **Integrations** > **Integrations**.
2. In the row for the **Opportunity details from Sales Cloud** sample, click the 
   icon, then click **Activate** when prompted.
3. Wait for the icon to turn green and the word **Active** to appear in the **Status** column, indicating that the integration is activated.

How To Run

1. Click the 
   icon to show a message with details about running, testing, and tracking the integration.
2. Enter the URL in a browser with a valid opportunity ID value. You can get the URL from the **Metadata URL** field in the message.

What Results Do You See

You receive the opportunity details response in your browser from Oracle Engagement Cloud.

How To Monitor

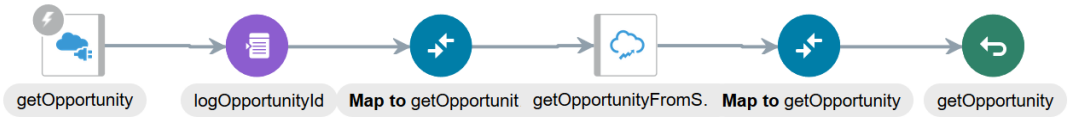
1. In the left navigation pane, click **Home** > **Monitoring** > **Integrations** > **Dashboards**.
   By default, the Dashboard page displays overall system status, including the percentage of successful messages, total number of messages, total number of successful messages, and total number of failed messages. Details about currently used connections, currently activated integrations, and scheduled integrations are also provided. You can also view the activity stream and download diagnostic logs and incident reports.
2. Select **Activity Stream** from the **View** menu to view details about the opportunity ID.
3. In the left navigation pane, click **Tracking** and note that the **Opportunity details from Sales Cloud** integration instance is listed as completed.
4. In the left navigation pane, click **Integrations** and note that the message was successfully received and processed without any errors.

How To View

1. In the left navigation pane, click **Home** > **Integrations** > **Integrations**.
2. Click the **Opportunity details from Sales Cloud** integration.

   A read-only version of the integration is displayed for viewing. Because the integration is active, it cannot be edited.

3. View the flow of the integration:
   - A REST Adapter is configured as a trigger (inbound) connection in the integration. The REST Adapter is configured with a resource endpoint of `/opportunity/{id}` and a GET operation, and retrieves the opportunity. This REST Adapter is triggered when you specify the URL in How to Run.
   - A logging message is created and logged to the activity stream.
   - The Oracle CX Sales and B2B Service Adapter is invoked in the integration. The adapter is configured with an opportunity business object and a CRUD Get operation to get opportunity details from Oracle Engagement Cloud. The opportunity response is returned to you.



Learn More About The Features in this Sample

- Creating Orchestrated Integrations
- REST Adapter Capabilities
- Logging Messages with a Logger Action
- Getting Started with the Mapper
- Oracle CX Sales and B2B Service Adapter Capabilities

Load Newer Versions of the Sample Integrations

When your instance is upgraded, the sample integration versions included with your previous instance version remain untouched and are not automatically updated. Because of this, you do not automatically get any samples that may have been updated to newer versions. To obtain the latest sample integration versions, you must perform the following steps. These steps enable you to load new sample integrations that overwrite previous integration samples with the same name and version. Any sample integrations you may have previously deleted are also reloaded. However, if a sample integration is active or locked, it is not overwritten.

To load newer versions of the sample integrations:

1. In the left navigation pane, click **Home** > **Integrations**.
2. In the upper right corner, click the **?** icon.
3. Select **Get Samples**.
4. Click **Get**.

   The sample integrations are loaded in bulk. A message is displayed indicating that the sample integrations have been loaded successfully. Any active or locked integrations are not loaded.

> **Note:**
>
> Each sample integration is loaded as part of its own package. Therefore, if you delete one of the sample integrations and create a new integration with the same name and version, the integration is not overwritten as long as you do not include it in that package. In the left navigation pane, click **Home** > **Integrations** > **Integrations** > **Packages** to see the package names associated with each sample integration.

Use Recipes to Build Integrations

Oracle Integration includes a growing portfolio of prebuilt integration solutions, known as *recipes*, which 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 solutions.

For a listing of all the recipes included with Oracle Integration, see Recipes.

About Integrations

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

Oracle Integration provides native connectivity to Oracle Software as a Service (SaaS) applications, such as Oracle CX Sales and B2B Service Adapter, Oracle Fusion Cloud B2C Service, and so on. Oracle Integration *adapters* simplify connectivity by handling the underlying complexities of connecting to applications using industry-wide best practices. You only need to create a *connection* that provides minimal connectivity information for each system. Oracle Integration *lookups* map the different codes or terms used by the applications you are integrating to describe similar items (such as country or gender codes). Finally, the visual data mapper enables you to quickly create direct mappings between the trigger and invoke data structures. From the mapper, you can also access lookup tables and use standard XPath functions to map data between your applications.

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

About Integrations Concepts

The following topics describe each of the components required to create an end-to-end integration. Each integration includes connections and mappings. You can also include

lookups, which are reusable mappings for the different codes and terms used in your applications to describe the same item. You can also group integrations into packages.

Topics:

About Connections

Connections define information about the instances of each configuration you are integrating. Oracle Integration includes a set of predefined *adapters*, which are the types of applications on which you can base your connections, such as Oracle CX Sales and B2B Service Adapter, Oracle Eloqua Cloud Adapter, Oracle Service Cloud (RightNow) Adapter, and others. A connection is based on an adapter. For example, to create a connection to a specific Oracle Service Cloud application instance, you must select the Oracle Service Cloud (RightNow) Adapter and then specify the WSDL URL, security policy, and security credentials to connect to it.

Video

Connection Creation

You can create a connection based on any of the following adapters.

Adapter	For Information
Adobe Sign Adapter	*Using the Adobe Sign Adapter with Oracle Integration*
Apache Kafka Adapter	*Using the Apache Kafka Adapter with Oracle Integration*
AS2 Adapter	*Using the AS2 Adapter with Oracle Integration*
Automation Anywhere Adapter	*Using the Automation Anywhere Adapter with Oracle Integration*
Box Adapter	*Using the Box Adapter with Oracle Integration*
DocuSign Adapter	*Using the DocuSign Adapter with Oracle Integration*
eBay Marketplace Adapter	*Using the eBay Marketplace Adapter with Oracle Integration*
Eventbrite Adapter	*Using the Eventbrite Adapter with Oracle Integration*
Evernote Adapter	*Using the Evernote Adapter with Oracle Integration*
Facebook Adapter	*Using the Facebook Adapter with Oracle Integration*
File Adapter	*Using the File Adapter with Oracle Integration*
FTP Adapter	*Using the FTP Adapter with Oracle Integration*

Adapter	For Information
Google Gmail Adapter	*Using the Google Gmail Adapter with Oracle Integration*
Google Calendar Adapter	*Using the Google Calendar Adapter with Oracle Integration*
Google Tasks Adapter	*Using the Google Tasks Adapter with Oracle Integration*
HubSpot Adapter	*Using the HubSpot Adapter with Oracle Integration*
Jira Adapter	*Using the Jira Adapter with Oracle Integration*
IBM DB2 Adapter	*Using the IBM DB2 Adapter with Oracle Integration*
IBM MQ Series JMS Adapter	*Using the IBM MQ Series JMS Adapter with Oracle Integration*
LinkedIn Adapter	*Using the LinkedIn Adapter with Oracle Integration*
Magento Commerce Cloud Adapter	*Using the Magento Commerce Cloud Adapter with Oracle Integration*
Mailchimp Adapter	*Using the Mailchimp Adapter with Oracle Integration*
Marketo Adapter	*Using the Marketo Adapter with Oracle Integration*
Microsoft Office 365 Calendar Adapter	*Using the Microsoft Office 365 Calendar Adapter with Oracle Integration*
Microsoft Office 365 People Adapter	*Using the Microsoft Office 365 People Adapter with Oracle Integration*
Microsoft Office 365 Outlook Adapter	*Using the Microsoft Office 365 Outlook Adapter with Oracle Integration*
Microsoft SQL Server Adapter	*Using the Microsoft SQL Server Adapter with Oracle Integration*
MySQL Adapter	*Using the MySQL Adapter with Oracle Integration*
Oracle Advanced Queuing (AQ) Adapter	*Using the Oracle Advanced Queuing (AQ) Adapter with Oracle Integration*
Oracle Autonomous Data Warehouse Adapter	*Using the Oracle Autonomous Data Warehouse Adapter with Oracle Integration*
Oracle Autonomous Transaction Processing Adapter	*Using the Oracle Autonomous Transaction Processing Adapter with Oracle Integration*
Oracle Cloud Infrastructure Streaming Service Adapter	*Using the Oracle Cloud Infrastructure Streaming Service Adapter with Oracle Integration*
Oracle Commerce Cloud Adapter	*Using the Oracle Commerce Cloud Adapter with Oracle Integration*
Oracle CPQ Adapter	*Using the Oracle CPQ Adapter with Oracle Integration*
Oracle CX Sales and B2B Service Adapter	*Using the Oracle CX Sales and B2B Service Adapter with Oracle Integration*
Oracle Database Adapter	*Using the Oracle Database Adapter with Oracle Integration*
Oracle Database Cloud Service Adapter	*Using the Oracle Database Cloud Service Adapter with Oracle Integration*
Oracle E-Business Suite Adapter	*Using the Oracle E-Business Suite Adapter with Oracle Integration*
Oracle Eloqua Cloud Adapter	*Using the Oracle Eloqua Cloud Adapter with Oracle Integration*
Oracle Enterprise Performance Management Cloud Adapter	*Using the Oracle Enterprise Performance Management Cloud Adapter with Oracle Integration*

Adapter	For Information
Oracle ERP Cloud Adapter	*Using the Oracle ERP Cloud Adapter with Oracle Integration*
Oracle Field Service Cloud Adapter	*Using the Oracle Field Service Cloud Adapter with Oracle Integration*
Oracle HCM Cloud Adapter	*Using the Oracle HCM Cloud Adapter with Oracle Integration*
Oracle Hospitality Adapter	*Using the Oracle Hospitality Adapter with Oracle Integration*
Oracle Intelligent Advisor Adapter	*Using the Oracle Intelligent Advisor Adapter with Oracle Integration*
Oracle Intelligent Track and Trace Adapter	*Using the Oracle Intelligent Track and Trace Adapter with Oracle Integration*
Oracle JD Edwards EnterpriseOne Adapter	*Using the Oracle JD Edwards EnterpriseOne Adapter with Oracle Integration*
Oracle Logistics Adapter	*Using the Oracle Logistics Adapter with Oracle Integration*
Oracle NetSuite Adapter	*Using the Oracle NetSuite Adapter with Oracle Integration*
Oracle Responsys Adapter	*Using the Oracle Responsys Adapter with Oracle Integration*
Oracle Service Cloud (RightNow) Adapter	*Using the Oracle Service Cloud (RightNow) Adapter with Oracle Integration*
Oracle Siebel Adapter	*Using the Oracle Siebel Adapter with Oracle Integration*
Oracle SOA Suite Adapter	*Using the Oracle SOA Suite Adapter with Oracle Integration*
Oracle Taleo Enterprise Edition Adapter	*Using the Oracle Taleo Enterprise Edition Adapter with Oracle Integration*
Oracle Taleo Business Edition (TBE) Adapter	*Using the Oracle Taleo Business Edition Adapter with Oracle Integration*
Oracle Unity Adapter	*Using the Oracle Unity Adapter with Oracle Integration*
Oracle Utilities Adapter	*Using the Oracle Utilities Adapter with Oracle Integration*
Oracle WebLogic JMS Adapter	*Using the Oracle WebLogic JMS Adapter with Oracle Integration*
PayPal Adapter	*Using the PayPal Adapter with Oracle Integration*
QuickBooks Adapter	*Using the QuickBooks Adapter with Oracle Integration*
REST Adapter	*Using the REST Adapter with Oracle Integration*
Salesforce Adapter	*Using the Salesforce Adapter with Oracle Integration*
Salesforce Commerce Cloud Adapter	*Using the Salesforce Commerce Cloud Adapter with Oracle Integration*
SAP Adapter	*Using the SAP Adapter with Oracle Integration*
SAP Ariba Adapter	*Using the SAP Ariba Adapter with Oracle Integration*
SAP ASE (Sybase) Adapter	*Using the SAP ASE (Sybase) Adapter with Oracle Integration*
SAP Commerce Cloud (Hybris) Adapter	*Using the SAP Commerce Cloud (Hybris) Adapter with Oracle Integration*

Adapter	For Information
SAP Concur Adapter	*Using the SAP Concur Adapter with Oracle Integration*
SAP SuccessFactors Adapter	*Using the SAP SuccessFactors Adapter with Oracle Integration*
SAP S/4HANA Cloud Adapter	*Using the SAP S/4HANA Cloud Adapter with Oracle Integration*
ServiceNow Adapter	*Using the ServiceNow Adapter with Oracle Integration*
Shopify Adapter	*Using the Shopify Adapter with Oracle Integration*
Slack Adapter	*Using the Slack Adapter with Oracle Integration*
SOAP Adapter	*Using the SOAP Adapter with Oracle Integration*
SugarCRM Adapter	*Using the SugarCRM Adapter with Oracle Integration*
SurveyMonkey Adapter	*Using the SurveyMonkey Adapter with Oracle Integration*
Trello Adapter	*Using the Trello Adapter with Oracle Integration*
Twilio Adapter	*Using the Twilio Adapter with Oracle Integration*
Twitter Adapter	*Using the Twitter Adapter with Oracle Integration*
UiPath Robotic Process Automation Adapter	*Using the UiPath Robotic Process Automation Adapter with Oracle Integration*
Workday Adapter	*Using the Workday Adapter with Oracle Integration*
Zendesk Adapter	*Using the Zendesk Adapter with Oracle Integration*

Oracle Integration Messaging

Oracle Integration Messaging enables you to publish messages to and subscribe to messages from Oracle Integration.

You may have business use cases in which you need to synchronize objects between applications. For example:

- Create an object in one application that causes the object to be created in other applications. For example, create a new account in Oracle CX Sales and B2B Service Adapter, which causes the creation of an Oracle RightNow organization and an Oracle Eloqua account.
- Enable multiple applications to subscribe to Oracle Integration and register for updates.
- Add or remove subscribers without impacting other subscribers or producers.

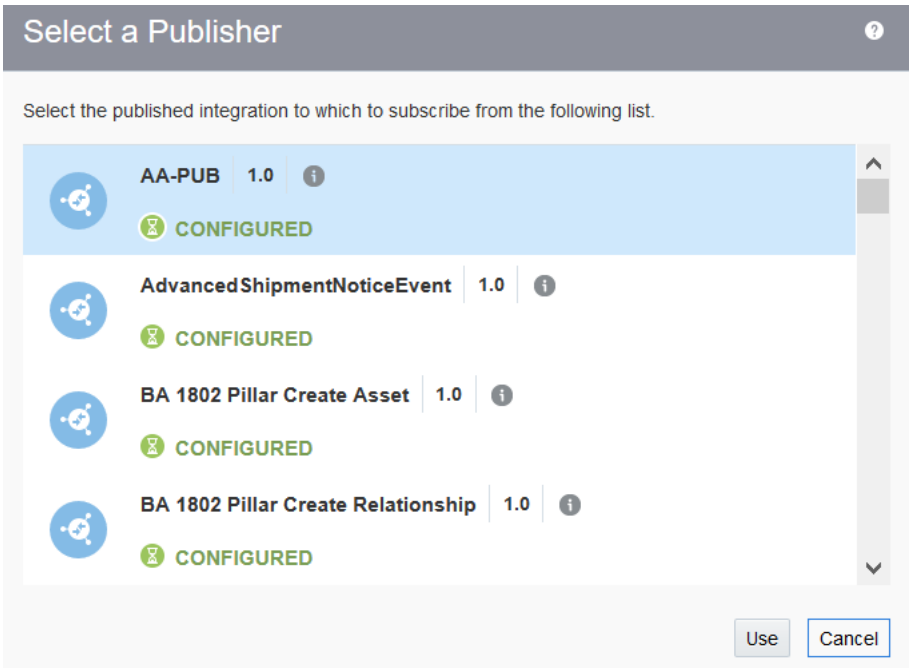
Oracle Integration Messaging addresses these business requirements through the creation of two types of integrations: one for publishing to Oracle Integration and one for subscribing to Oracle Integration.

- You create an integration that enables you to publish messages to Oracle Integration by selecting the **Publish to OIC** option in the Select Integration Style dialog. In this integration:
  - Oracle Integration is added as an invoke and is automatically configured.
  - You configure a trigger (source) adapter (for example, Oracle RightNow, Oracle CX Sales and B2B Service Adapter, or another).
  - The message to pass to Oracle Integration is opaque, so no request mapper support is provided.

- No trigger (source) enrichment mapper support is provided.
- Multiple publishers targeting a single message destination is not supported.

> **Note:**
>
> Modifying the publisher after creating the subscribers can potentially impact the subscribers. For example, if you change the published object, any existing subscriber mappings are impacted.

- You create an integration that enables you to subscribe to messages from Oracle Integration by selecting the **Subscribe to OIC** option in the Select Integration Style dialog. In this integration:
  - Oracle Integration is added as a trigger (source).
  - You are prompted to select the published integration to which to subscribe.



  - You configure an invoke adapter to subscribe to and receive messages from Oracle Integration.
  - Response mapper support is provided between the published object and the subscriber's application object.
  - Trigger (source) enrichment mapper support is provided.

See Create an Integration to Publish Messages to Oracle Integration and Create an Integration to Subscribe to Oracle Integration.

Related Topics

See the following sections for additional information.

- See Adding a Trigger Connection and Adding an Invoke Connection.

- See Create a Connection.

About Integrations in Oracle Integration

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

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

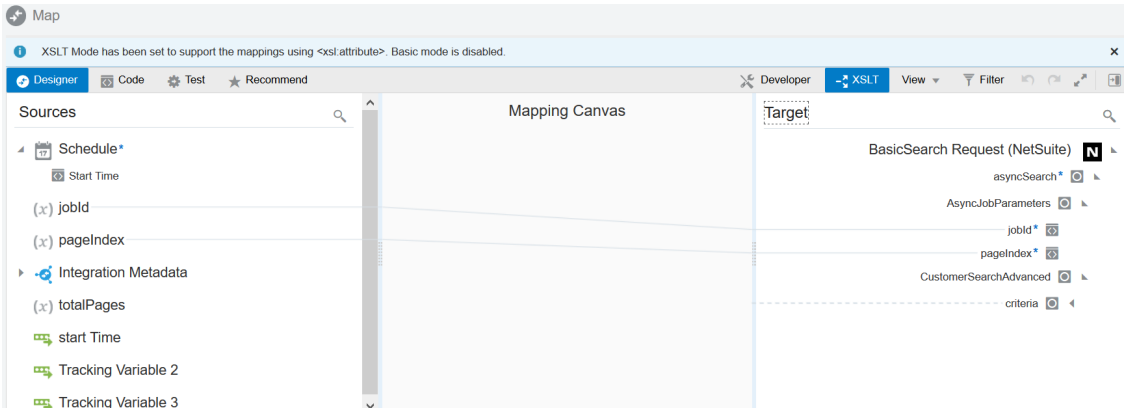
Video

Video

About Mappings

One of the key tasks to any integration is defining how data is transferred, or *mapped*, between two applications.

In most cases, the messages you want to transfer between the applications in an integration have different data structures. A visual mapper enables you to map element nodes between applications by dragging source element nodes onto target element nodes. When you open the mapper for a request or response message in an integration, the data structures are automatically populated with the information pulled from the source and target connections. You can expand and load data structure levels on demand to display additional levels. There is no limit on the levels of display.



The maps you create are called transformation maps, and use the eXtensible Stylesheet Language (XSL) to describe the data mappings, which let you perform complex data manipulation and transformation. A standard set of XSLT constructs are provided (for example, `xsl:if`, `xsl:for-each`, and others). A specialized function is also provided for you to reference lookups directly from the mapper.

> **Note:**
>
> The mapper supports XSL version 2.0. Version 1.0 is not supported.

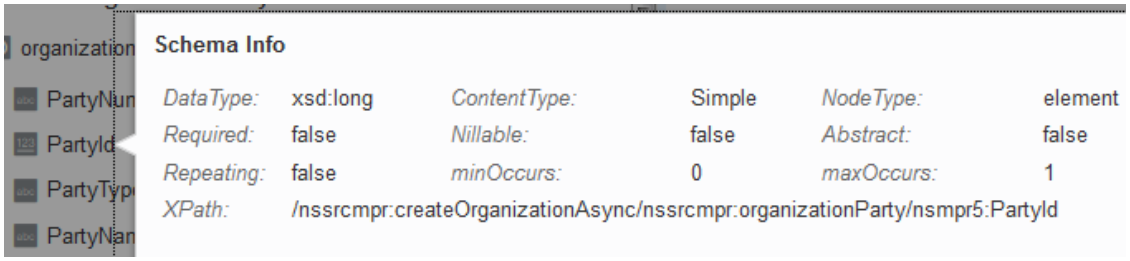
The mapper supports both qualified and unqualified schemas (that is, schemas without `elementFormDefault="qualified"`). Elements and attributes with and without namespace prefixes are also supported.

Substitution groups in schemas are supported. You can see all the substitutable elements in a base element in the mapper, and select the one to use.

Extended data types are also supported.

Elements and attributes for which mapping is required are identified by a blue asterisk (*) to the left of their names. To display only required fields, click the **Filter** icon in the mapper toolbar, select **Required Fields**, and click **Apply**.

You can also right-click elements and attributes and select **Node Info** to show specific schema details such as the data type, if mapping is required, and so on.



Additional custom annotations can also be displayed. These annotations are currently only available with the Oracle Sales Cloud Adapter. The Oracle CX Sales and B2B Service Adapter obtains this information from the applications and annotates it in the integration WSDL. This information is then read and made visible as annotations in the mapper (for example, title and description). This information can help you better understand what data is being mapped.

The mapper toolbar provides the following functionality.

Element	Description
Designer	Click to return to the mapping canvas when you are inside the Code, Test, or Recommend page.
Code	You can view the XSLT code being created as you design your mappings.
Test	Once you complete designing your mappings, you can test them by entering sample content of the message to process in the mapping tester.
Recommend	If you enable the recommendations engine, you can accept the target element recommendations of the engine when creating mappings. This eliminates the need to analyze and perform each individual source-to-target mapping.
Developer	Click to disable user-friendly, source and target element names in the mapper. By default, user-friendly element names are shown.

Element	Description
XSLT	Click to show the XSLT functions.
View	You can select the following options: • Select to show the namespace prefixes on source and target element nodes. • Select to show the types (prefixes and data types) on source and target element nodes.
Filter	You can filter the display of element nodes, error messages, and warnings in the source or target data structures.
	You can select to undo the previous action performed in the mapper. For example, if you perform a mapping, then press this button, the mapping is removed. The link is disabled when all actions have been undone.
	You can redo the action that was undone.
	You can maximize the size of the mapper. This is useful when working with large schemas.
	You can add functions, operators, and XSLT expressions to your mappings.

About Oracle Integration Enrichments

You may have business use cases in which you need to enhance data by calling another service *before* sending data to an invoke service or *before* sending data back to a requestor. To address this business requirement, you can optionally add enrichment data sources to the request part, the response part, or both parts of an integration. Enrichments participate in the overall integration flow and can be used in the request and/or response payloads between the trigger and invoke services. Enrichments subscribe to a synchronous request and response message pattern.

Enrichments enable you to:

- Add additional information. For example, your business use case may require you to:
  - Add a stock price
  - Increase on-site quantities of a product
  - Estimate local currency
- Convert data, such as mapping data between account numbers. The ability to map data between the request/response payload and the enrichment source application is a key feature of enrichments.

See Add Request and Response Enrichments.

About Oracle Integration Lookups

Use lookups in your integrations to create reusable tables that map the different terms used to describe the same item across your applications.

A lookup associates values used by one application for a specific item to the values used by other applications for the same item. For example, one application uses a specific set of codes to describe countries, while another application uses a different set of codes to describe the

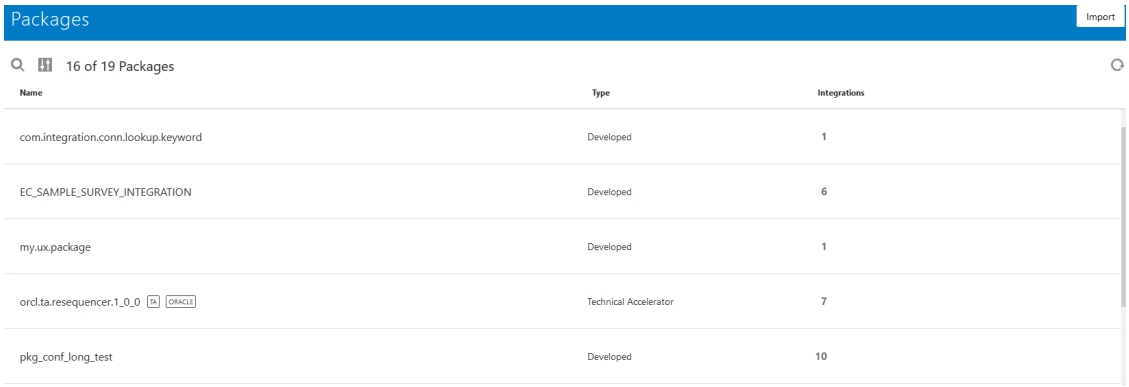
same countries. Lookups can be used for items such as mapping gender codes, nationality codes, currency codes—any type of information that your applications must share with each other but that they represent differently. You may have several lookups for one integration, depending on the number of fields that require mapping. Lookups are also reusable, and can be used by multiple integrations. Lookups are based on a static definition, meaning you create and populate them during design time, and are not changed by runtime activities. These tables are used for looking up values only.

See Map Data and Create Lookups.

About Oracle Integration Packages

You can group one or more integrations into a single structure called a package. Packages enable you to easily import and export a group of integrations to and from Oracle Integration. You can import packages from the Oracle Marketplace. These packages consist of a series of prebuilt integrations provided by Oracle. You can also import and export packages that consist of integrations that you or other users created. Packages are optional, meaning that integrations do not need to be part of packages. However, for a package to exist, it must include at least one integration. Packages cannot be locked to exclude other users of your Oracle Integration instance.

Packages are displayed on the Packages page in Oracle Integration. From this page, you can view, delete, import, and export packages. You create packages when you create an integration in the Create Integration dialog. You can also update an integration's package in the Update Integration dialog.



See Manage Packages and Create Integrations.

About Creating Hybrid Integrations Using Oracle Integration

A lot of business use cases require integration between applications hosted on public cloud and resources residing in an on-premises network or private cloud. For example, consider a business case where a quote or sales order configured through an Oracle Configure Price Quote application has to be sent to an Oracle E-Business Suite application, hosted in an on-premise network, for creation and fulfillment of the sales order. To facilitate such hybrid integrations, Oracle Integration provides the necessary infrastructure as well as architecture patterns.

This kind of hybrid integration enables you to have flows hosted on Oracle Integration that:

- Access SOAP/REST endpoints exposed by applications such as Oracle E-Business Suite, Siebel, and JD Edwards and any on-premises home grown SOAP/REST APIs

- Access non-HTTP-based endpoints such as databases, JMS, AQ, local file systems, SAP, and others

The above capabilities enable you to implement use cases such as the following:

- Send requests from a cloud application (for example, send a create service order request from an Oracle Service Cloud application) to an on-premises E-Business Suite application
- Synchronize bulk data extracts of a product from a product data hub in Oracle ERP Cloud with an on-premises Oracle database or an Oracle Database Cloud Service instance using the connectivity agent
- Synchronize customers that are added/updated in an on-premises SAP application with SaaS applications such as Oracle CX Sales and B2B Service Adapter, Oracle CPQ, Oracle Service Cloud, and Salesforce.com

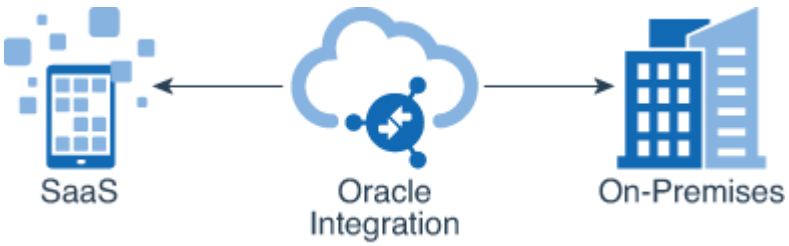
Oracle Integration provides a component called the *connectivity agent* to facilitate hybrid integrations. See About the Connectivity Agent.

For different connection patterns you can use to create hybrid integrations, see Connection Patterns for Hybrid Integrations.

About the Connectivity Agent

Using the connectivity agent, you can create hybrid integrations and exchange messages between applications in private or on-premises networks and Oracle Integration. Message payloads of up to 10 MB are supported through the use of compression and files or attachments up to 1 GB can be transferred back and forth between Oracle Integration and on-premises sFTP servers. Similarly, attachments up to 1 GB can be sent to on-premises SOAP/REST endpoints using the SOAP or REST adapter. The connectivity agent also provides multithreading support, which allows for multiple executors to perform downstream message processing.

In addition, the connectivity agent framework enables SaaS applications in the cloud to interact through Oracle Integration with on-premises systems



Connectivity Agent Components

The connectivity agent consists of the following components:

- SaaS agent: This agent is installed and runs in Oracle Integration and supports communication with on-premises applications. There is one SaaS agent per Oracle Integration environment.
- On-premises agent: This agent is installed and runs in an on-premises environment on the same network as internal systems such as Oracle E-Business Suite, Oracle Siebel, Oracle Database, and others. You download the on-premises agent installer from the Agents page in Oracle Integration to your on-premises environment for installation. There can be multiple host systems, each running one or more agents, in a cloud/on premises topology.

The on-premises agent does not permit any explicit inbound connections. All connections are established from the on-premises environment to Oracle Integration.

Connectivity Agent Functionality

The connectivity agent provides the following functionality:

> **Note:**
>
> While multiple connectivity agents can run on a single host, this is *not* the recommended practice. If you follow this practice, you must ensure that the physical host has enough resources to run multiple connectivity agents.

- No ports are opened on the on-premises system for communication.
- All communication is secured using SSL.
- The on-premises connectivity agent registers with Oracle Integration over SSL using the provided Oracle Integration credentials.
- The on-premises connectivity agent checks for work by making outbound requests through the firewall.
- The on-premises connectivity agent can use a proxy to access the internet (the same proxy as other internal applications and browsers use). Authentication support for outbound proxy access is provided.
- The on-premises connectivity agent connections are configured by the agent retrieving the configuration details from Oracle Integration.
- The on-premises connectivity agent processes requests by pulling messages from Oracle Integration across SSL.
- The on-premises connectivity agent posts responses by pushing messages to Oracle Integration across SSL.
- All communication is initiated by the on-premises connectivity agent.
- No private SOAP-based web services are exposed.
- No existing J2EE container is required to deploy the on-premises connectivity agent.
- No data is persisted in the on-premises agent.

Adapter Connections that Work with the Connectivity Agent

The on-premises agent works with the following adapter connections.

- Outbound (invoke) adapters: The following adapters can be configured as invoke connections in an integration to support communication with endpoint applications:
  - Apache Kafka Adapter
  - File Adapter
  - FTP Adapter
  - IBM DB2 Adapter
  - IBM MQ Series JMS Adapter
  - Microsoft SQL Server Adapter
  - MySQL Adapter

  - Oracle Advanced Queuing (AQ) Adapter
  - Oracle Autonomous Data Warehouse Adapter
  - Oracle Autonomous Transaction Processing Adapter
  - Oracle Database Adapter
  - Oracle Database Cloud Service Adapter
  - Oracle E-Business Suite Adapter
  - Oracle JD Edwards EnterpriseOne Adapter
  - Oracle Siebel Adapter
  - Oracle SOA Suite Adapter
  - Oracle WebLogic JMS Adapter
  - REST Adapter
  - SAP Adapter
  - SAP ASE (Sybase) Adapter
  - SOAP Adapter
- Inbound (trigger) adapters: The following adapters can be configured as trigger connections in an integration:
  - Apache Kafka Adapter
  - File Adapter
  - IBM DB2 Adapter
  - IBM MQ Series JMS Adapter
  - Microsoft SQL Server Adapter
  - MySQL Adapter
  - Oracle Advanced Queuing (AQ) Adapter
  - Oracle Autonomous Data Warehouse Adapter
  - Oracle Autonomous Transaction Processing Adapter
  - Oracle Database Adapter
  - Oracle Database Cloud Service Adapter
  - Oracle E-Business Suite Adapter
  - Oracle JD Edwards EnterpriseOne Adapter
  - Oracle Siebel Adapter
  - JMSOracle WebLogic JMS Adapter
  - SAP Adapter
  - SAP ASE (Sybase) Adapter

Connection Patterns for Hybrid Integrations

Use the connectivity agent in any of the following patterns to set up a connection between an application on your private (or on-premises) network and Oracle Integration.

You can set up a connection over the public internet or choose to configure an exclusive connection using FastConnect, which provides a faster, more reliable networking experience

compared to the internet. You'll use the connectivity agent to communicate with Oracle Integration irrespective of the connection pattern you choose; employing FastConnect only ensures that the traffic between your private (on-premises) network and Oracle Integration doesn't go over the public internet and remains private.

The patterns you can use are listed here:

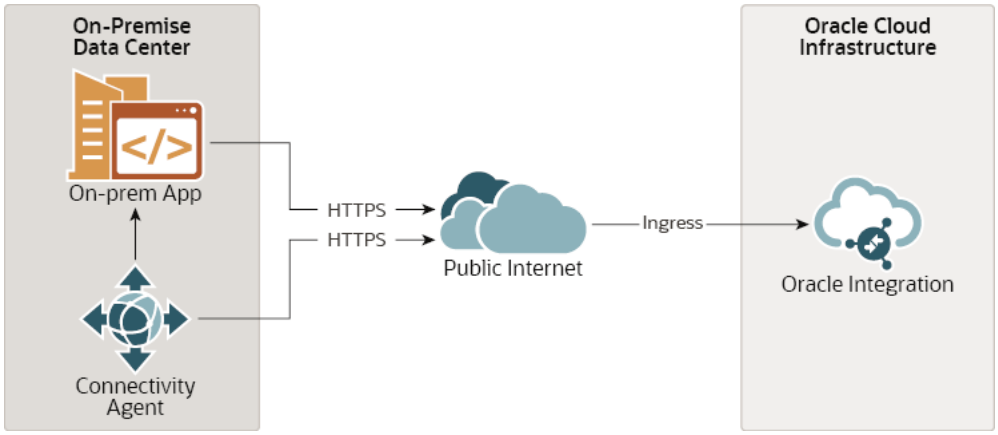
- Public Internet Pattern
- Oracle Cloud Infrastructure Only - Virtual Cloud Network Pattern
- FastConnect Public Peering Pattern
- FastConnect Private Peering Pattern

Public Internet Pattern

Install the connectivity agent on your private (on-premises) network. The inbound and outbound traffic to Oracle Integration goes over the public internet. For the outbound traffic from Oracle Integration, the connectivity agent initiates a secure connection to Oracle Integration, retrieves the request, and then invokes the required API in the on-premises application.

When you employ the connectivity agent, you don't have to open firewalls to access applications on your private network. In addition, all messages between the private network and Oracle Integration are encrypted.

To install and configure the connectivity agent, see Manage the Agent Group and the On-Premises Connectivity Agent.



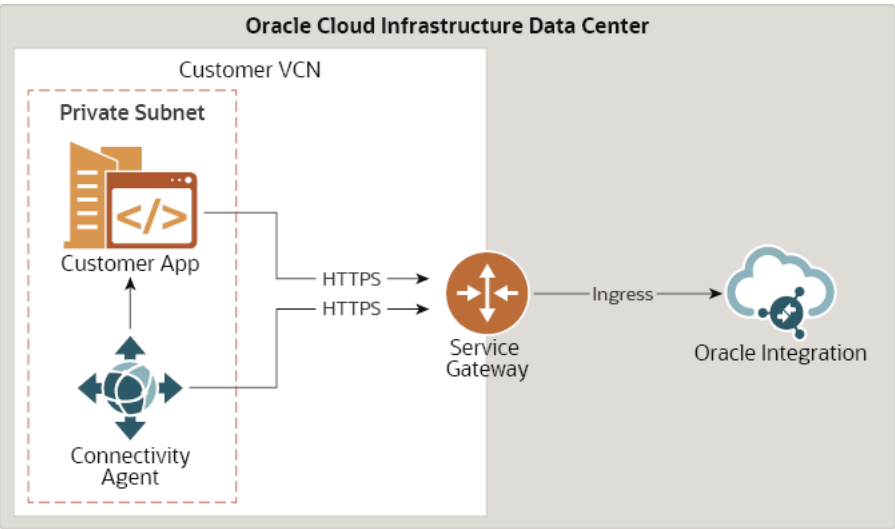
Oracle Cloud Infrastructure Only - Virtual Cloud Network Pattern

Install the connectivity agent in your Virtual Cloud Network (VCN) within Oracle Cloud Infrastructure, and configure a service gateway to route the traffic from the VCN to Oracle Integration. Use this pattern if you have applications, such as Oracle E-Business Suite, running in a private subnet within Oracle Cloud Infrastructure. In this case, all traffic is routed locally and the public internet is not involved.

While not strictly required, it is highly recommended that all access from the VCN go through the service gateway. Service gateways primarily ensure that access to Oracle-hosted services is routed over the internal network. Users are not charged for service gateways. Service gateways only work within a region, and not across regions. For access across regions, traffic is still routed through a NAT gateway.

A service gateway is a common configuration for users that have implemented FastConnect or VPN/IPsec private peering to route traffic to Oracle Integration (ingress), including the connectivity agent in a private subnet.

For details on configuring a service gateway, see FastConnect and VPN with Oracle Integration Cloud (OIC).



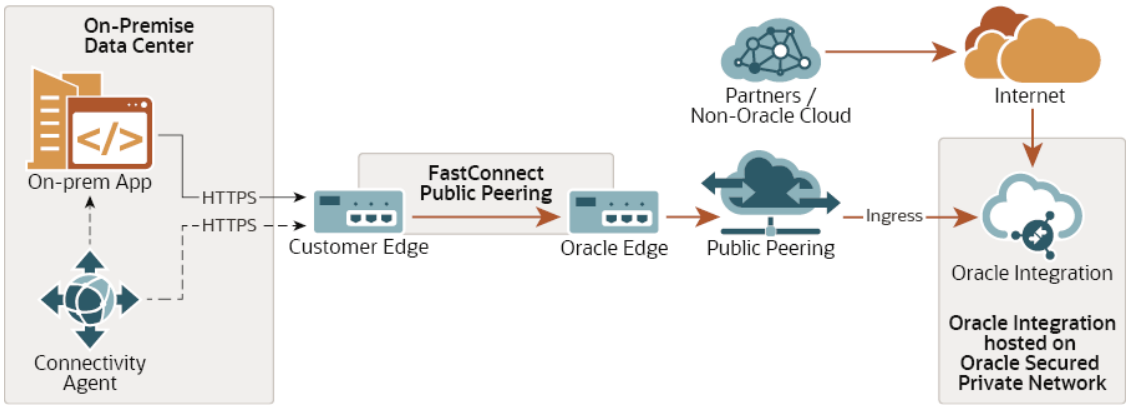
FastConnect Public Peering Pattern

Install the connectivity agent on your private (on-premises) network, and set up an exclusive connection between your network and Oracle Integration using a FastConnect public peering link. The inbound and outbound traffic to Oracle Integration goes through the FastConnect link. This connection pattern provides a faster and more reliable networking experience compared to the public internet pattern.

Follow the steps listed here to configure this pattern:

1. Subscribe to FastConnect with the public peering option. Currently, Oracle Integration directly supports only public peering with FastConnect. If you want to use the private peering option, you'll need to additionally use a VCN and a service gateway. See the FastConnect Private Peering Pattern.
   For detailed information on requirements and best practices for setting up an Oracle Cloud Infrastructure FastConnect, see FastConnect.
2. Configure your private (on-premises) network to route traffic through FastConnect.
   The FastConnect link contains the public IP addresses of Oracle Integration.
3. Finally, configure the connectivity agent to handle the outbound traffic from Oracle Integration to the on-premises application.
   The connectivity agent also acts as a client to FastConnect and uses public peering.

> **Note:**
>
> With FastConnect public peering, you must deploy the connectivity agent on your private or on-premises data center.



FastConnect Private Peering Patterns

In addition to providing fast and reliable connectivity, the FastConnect private peering patterns provide additional security to prevent traffic analysis.

> **Note:**
>
> The private peering patterns also apply to Virtual Private Networks (VPN) and are identical except that the FastConnect private peering link is replaced with a VPN.

The connection patterns are as follows:

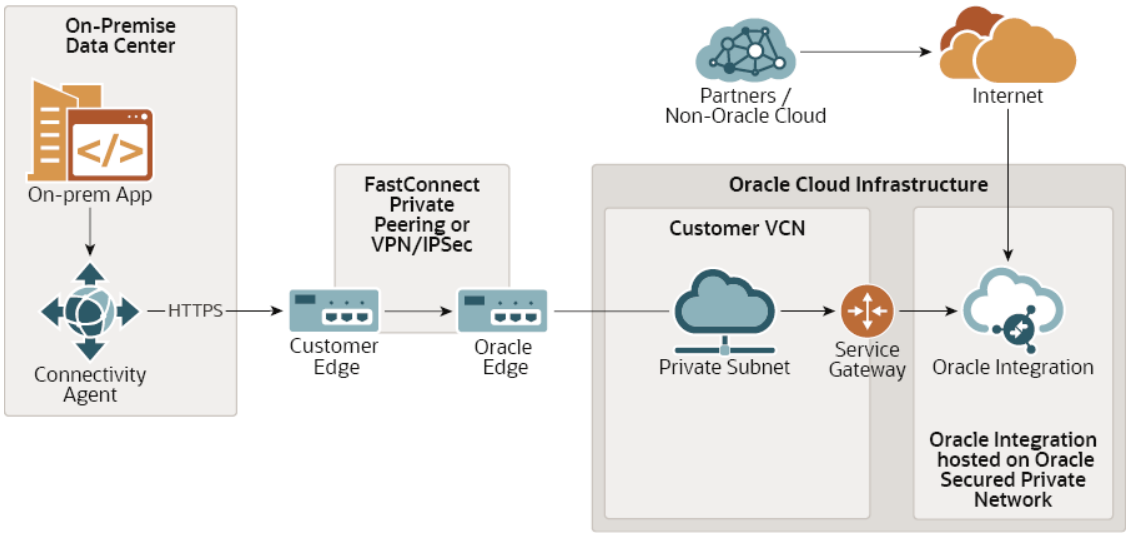
- Connectivity Agent Deployed in Private or On-Premise Network
- Connectivity Agent Deployed in VCN

Connectivity Agent Deployed in Private or On-Premises Network

Install the connectivity agent on your private (on-premises) network, and set up a private connection between your network and VCN using FastConnect private peering or VPN. In addition, configure a service gateway to route the traffic from the VCN to Oracle Integration. To use FastConnect, you should first subscribe to FastConnect with the private peering option. See FastConnect. The FastConnect link contains the private IP addresses of the VCN.

If you want use VPN, see VPN Connect.

To configure a service gateway, see FastConnect and VPN with Oracle Integration Cloud (OIC).

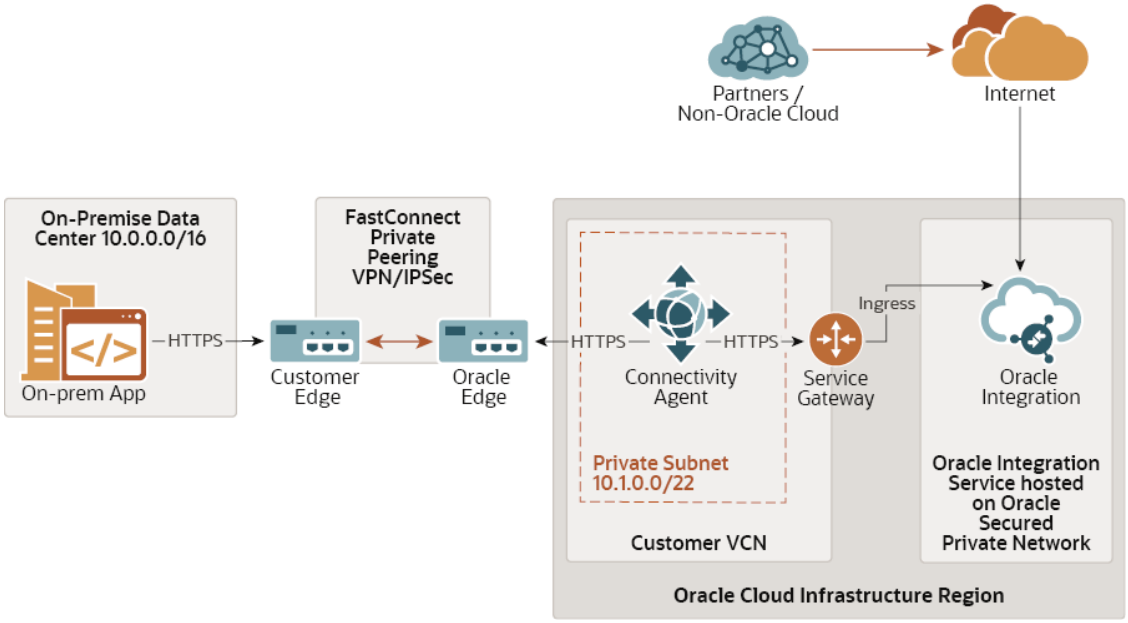


Connectivity Agent Deployed in VCN

Install the connectivity agent in your VCN within Oracle Cloud Infrastructure, and set up a private connection between your network and the connectivity agent using FastConnect private peering or VPN. In addition, configure a service gateway to route the traffic from the VCN to Oracle Integration. You can use this pattern if you have limited capacity or resource constraints on your data center.

For details on configuring a service gateway, see FastConnect and VPN with Oracle Integration Cloud (OIC).

> **Note:**
>
> The connectivity agent deployed in VCN can also be used to access resources deployed in Oracle Cloud Infrastructure VCN.



Workflow for Using the Connectivity Agent

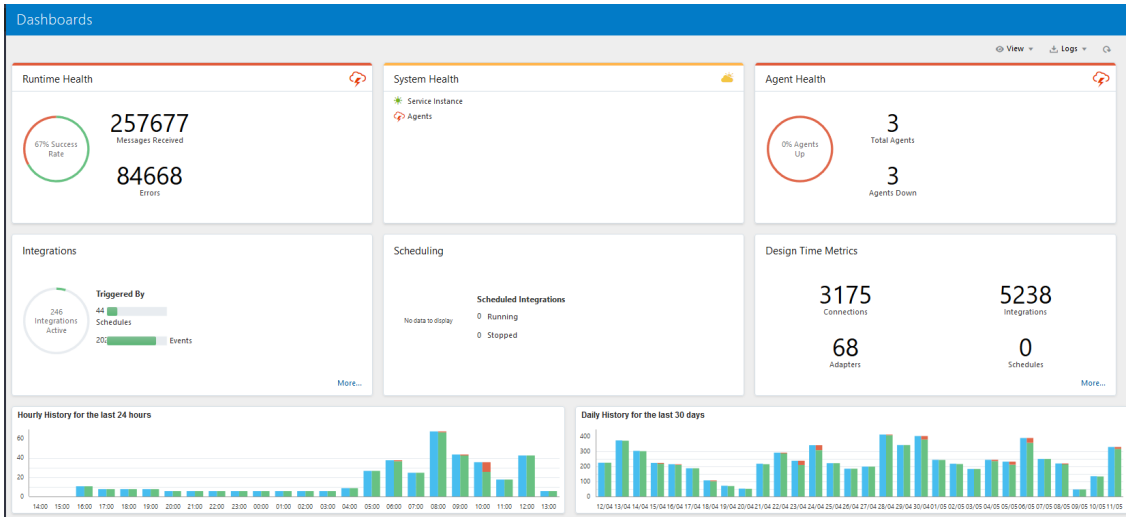
Follow this workflow to use the connectivity on-premises agent.

Task	Documentation
Create a connectivity agent group.	Create an Agent Group
Download and run the on-premises connectivity agent installer on your host. During installation setup, you associate the on-premises connectivity agent with the agent group.	Download and Run the Connectivity Agent Installer
Create an adapter connection in Oracle Integration and associate the connection with the connectivity agent group.	Create Connections
Design an integration that uses this connection.	Understand Integration Creation and Best Practices
Activate the integration.	Activate an Integration

About Monitoring

The Oracle Integration dashboard displays information about the current state of all your running integrations.

The dashboard gives you a quick view into the performance metrics for all of your active integrations. The graph includes the total number of messages processed, the average processing time, the number of messages with errors, and the success rate. The Integrations tab lists metrics for each running integration, and you can view a graph of the metrics for each integration in the list. The dashboard also provides a view of recent activity for running integrations and the ability to download all activities.



About Error Management

You can manage integration errors from the Errors pages in Oracle Integration. The Errors pages display information about individual integration instances and group the errors by integrations and connections over a specified time period.

You can perform the following tasks from the Errors page and its subpages:

- Search for and display errors by integration name or the total error count over a specific time period
- Resubmit errors
- Discard (remove) errors by integration name
- Display errors by connection name over a specific time period
- Discard errors by connection name
- Display errors by integration instance identifier, error location, or time of occurrence over a specific time period
- View the instance in which errors occurred
- View the error message
- View and discard errors by instance ID
- View the activity stream and message payload of a failed integration instance
- View the business identifiers of a failed integration instance

Integration Failure Scenarios

Integrations can fail for the following reasons:

- A call to a target system fails because the target system is down (for a short or long time period).
- A source-to-target transformation or target-to-source transformation fails because of faulty XSL coding, an invalid lookup call, or other system issue.
- A target system call results in a business failure for the following possible reasons:
  - Incorrect target application configuration
  - Invalid lookup data
  - Invalid business data
- A call to a trigger system (with a response message) fails because the trigger system is down.
- An enrichment step fails.
- A publish/subscribe scenario fails.
- Any other Oracle Integration system failure.

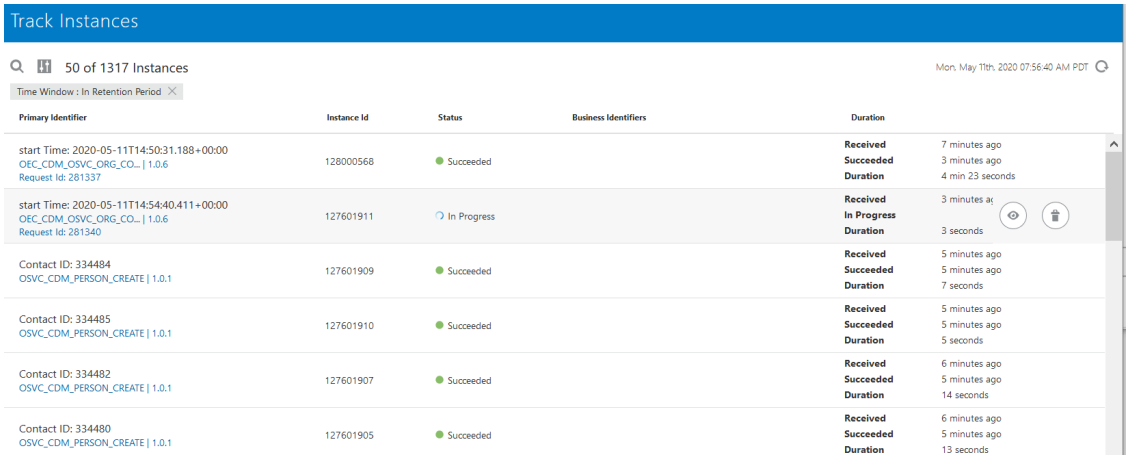
See Manage Errors.

About Business Identifiers for Tracking Fields in Messages

Business identifiers enable you to track payload fields in messages during runtime. You define up to three business identifiers on payload fields during design time. You designate one field as

the primary business identifier field, which enables message fields to be tracked during runtime.

During runtime, the Track Instances page displays information about the status of business identifiers and their values in your integrations.



If you have defined business identifiers in integrations that have failed, you can view details on the Errors page.

See Assign Business Identifiers for Tracking Fields in Messages, Manage Business Identifiers for Tracking Fields in Messages, and Manage Errors.

Start Oracle Integration

To get started with Oracle Integration, you must have a user account already set up. You access Oracle Integration through a web browser.

You receive your user account when you subscribe to Oracle Integration.

To access Oracle Integration:

1. In a web browser, enter the URL of your instance that you received from your administrator:

   `https://instance_URL:/ic/home/`

2. On the login page, enter your user name and password.

   Once you sign in, the Home page is displayed. Select any item on this page to learn more about Oracle Integration and its features.

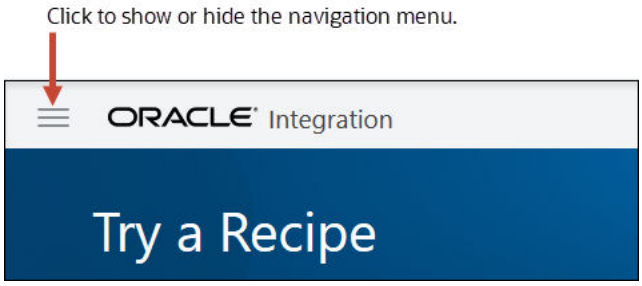
3. Click **Home**.

   See What You Can Do on the Home Page.

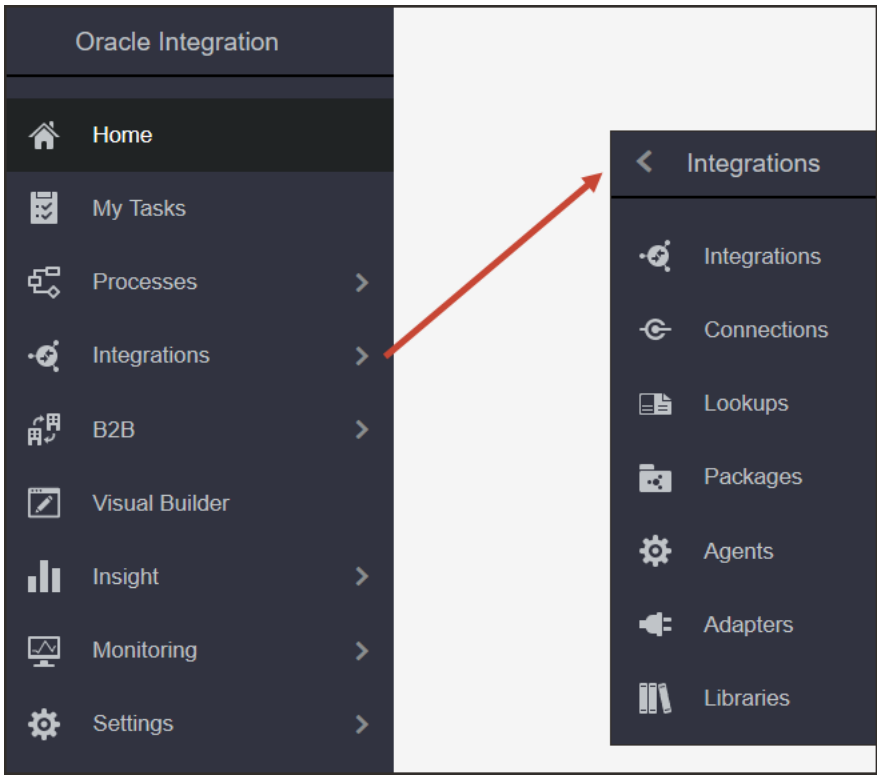
Navigate Oracle Integration

Use the navigation pane to get access to all the pages, tools, dashboards, and settings in Oracle Integration.

When you start Oracle Integration, the navigation pane is hidden. To view the navigation pane, click the **Navigation Menu** icon in the top-left corner of the Home page.



The menu displayed in the navigation pane depends on your assigned role, your selection, and your location in Oracle Integration. The navigation pane may display the main menu or a sub-menu, such as the one for Integrations.



Take a moment to learn about the main menu. It gives you access to the Oracle Integration runtime environment, design-time environment, development tools, and administration settings.

Click...	To...
Home	Access the launch pad and high-level dashboard for Oracle Integration, search for accelerators and recipes, and open recently worked on projects.
My Tasks	Launch the runtime environment for process tasks. Start process applications, work on tasks as a user, view dashboards, and perform runtime administration such as mapping process roles. See Tour the My Tasks Page in *Using Processes in Oracle Integration Generation 2*.

Click...	To...
 **Processes**	Launch the design-time environment for process applications. Create process applications from scratch or by using QuickStart Apps and samples. Model structured and dynamic processes in the process editor, and create web forms and decision models. Perform design-time administration such as playing, testing, and deploying applications. See Tour of the Process Applications Page in *Using Processes in Oracle Integration Generation 2*.
 **Integrations**	Launch the design-time environment for integrations. Configure connections, and create and activate integrations. Work with lookups, packages, agents, and adapters. See Get Started with Integrations in *Using Integrations in Oracle Integration Generation 2*.
 **B2B**	Exchange business documents between Oracle Integration and a trading partner securely and reliably. See Introduction to B2B for Oracle Integration in *Using B2B for Oracle Integration Generation 2*.
 **Visual Builder**	Create and publish web and mobile applications. See Get Started with Visual Builder in *Developing Applications with Oracle Visual Builder in Oracle Integration Generation 2*.
 **Insight**	Model business processes and monitor business metrics in real-time using interactive dashboards to take timely actions. See Introduction to Insight in *Using Integration Insight in Oracle Integration Generation 2*.
 **Monitoring**	Monitor integrations in the runtime environment. View the dashboard to see the performance of your integrations, track business identifiers associated with your integrations, monitor scheduled integration runs, or view and manage errors. See View the Dashboard in *Using Integrations in Oracle Integration Generation 2*.
 **Settings**	Configure settings for error logs and manage security certificates. Specify other settings such as notification services and mapping recommendations. See Configure Settings for Error Logs in *Administering Oracle Integration Classic* and Upload an SSL Certificate in *Using Integrations in Oracle Integration Generation 2*.

Typical Workflow for Creating Integrations with Oracle Integration

You follow a very simple workflow to develop integrations in Oracle Integration. The only prerequisites for creating an integration are that the application connections you need are in place and that any lookups you want to use to map information between applications are created.

Step	Description	More Information
1	Create the 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 Connection

Step	Description	More Information
2	Install the on-premises connectivity agent if you need to create integrations and exchange messages between on-premises applications and Oracle Integration.	Manage the Agent Group and the On-Premises Connectivity Agent
3	(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).	Create a Lookup
4	Create the integration. When you do this, you add trigger and invoke connections, assign business identifiers, and then map the data between the two.	1. Create Integrations 2. Understand Trigger and Invoke Connections 3. Assign Business Identifiers 4. Map Data in *Using the Oracle Mapper*
5	Activate the integration.	Activate an Integration
6	Monitor the integration on the dashboard.	View the Dashboard
7	Track payload fields in messages during runtime.	Assign Business Identifiers and Track Business Identifiers in Integrations During Runtime
8	Manage errors at the integration level, connection level, or specific integration instance level.	Manage Errors

2

Create Connections

You define connections to the specific cloud applications that you want to integrate. The following topics describe how to define connections:

Topics:

Inbound Endpoints and Transport Layer Security Server Version 1.2 Support

All inbound endpoints for Oracle Integration integrations are hosted on SSL servers that can accept requests coming from clients supporting transport layer security (TLS) 1.2. This is true regardless of whether they are SOAP- or REST-enabled and regardless of the adapter used as the trigger connection. TLS version 1.3 is *not* supported for either inbound integrations or outbound integrations.

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 and Security Properties

Enter connection and security property information for the adapter.

Configure Connection Properties

Enter connection information so your application can process requests.

1. Go to the **Connection Properties** section.
2. See the following sections for information about specifying connection properties for the adapter you selected.

Adapter	For Information
Adobe Sign Adapter	*Using the Adobe Sign Adapter with Oracle Integration*
Apache Kafka Adapter	*Using the Apache Kafka Adapter with Oracle Integration*
AS2 Adapter	*Using the AS2 Adapter with Oracle Integration*
Automation Anywhere Adapter	*Using the Automation Anywhere Adapter with Oracle Integration*
Box Adapter	N/A
DocuSign Adapter	*Using the DocuSign Adapter with Oracle Integration*
eBay Marketplace Adapter	*Using the eBay Marketplace Adapter with Oracle Integration*
Eventbrite Adapter	*Using the Eventbrite Adapter with Oracle Integration*
Evernote Adapter	*Using the Evernote Adapter with Oracle Integration*
Facebook Adapter	*Using the Facebook Adapter with Oracle Integration*
File Adapter	*Using the File Adapter with Oracle Integration*
FTP Adapter	*Using the FTP Adapter with Oracle Integration*
Google Gmail Adapter	*Using the Google Gmail Adapter with Oracle Integration*
Google Calendar Adapter	*Using the Google Calendar Adapter with Oracle Integration*
Google Tasks Adapter	*Using the Google Tasks Adapter with Oracle Integration*
HubSpot Adapter	*Using the HubSpot Adapter with Oracle Integration*
Jira Adapter	*Using the Jira Adapter with Oracle Integration*
IBM DB2 Adapter	*Using the IBM DB2 Adapter with Oracle Integration*
IBM MQ Series JMS Adapter	*Using the IBM MQ Series JMS Adapter with Oracle Integration*
LinkedIn Adapter	*Using the LinkedIn Adapter with Oracle Integration*
Magento Commerce Cloud Adapter	*Using the Magento Commerce Cloud Adapter with Oracle Integration*
Mailchimp Adapter	*Using the Mailchimp Adapter with Oracle Integration*
Marketo Adapter	*Using the Marketo Adapter with Oracle Integration*
Microsoft Office 365 Calendar Adapter	*Using the Microsoft Office 365 Calendar Adapter with Oracle Integration*
Microsoft Office 365 People Adapter	*Using the Microsoft Office 365 People Adapter with Oracle Integration*
Microsoft Office 365 Outlook Adapter	*Using the Microsoft Office 365 Outlook Adapter with Oracle Integration*
Microsoft SQL Server Adapter	*Using the Microsoft SQL Server Adapter with Oracle Integration*
MySQL Adapter	*Using the MySQL Adapter with Oracle Integration*

Adapter	For Information
Oracle Advanced Queuing (AQ) Adapter	*Using the Oracle Advanced Queuing (AQ) Adapter with Oracle Integration*
Oracle Autonomous Data Warehouse Adapter	*Using the Oracle Autonomous Data Warehouse Adapter with Oracle Integration*
Oracle Autonomous Transaction Processing Adapter	*Using the Oracle Autonomous Transaction Processing Adapter with Oracle Integration*
Oracle Cloud Infrastructure Streaming Service Adapter	*Using the Oracle Cloud Infrastructure Streaming Service Adapter with Oracle Integration*
Oracle Commerce Cloud Adapter	*Using the Oracle Commerce Cloud Adapter with Oracle Integration*
Oracle CPQ Adapter	*Using the Oracle CPQ Adapter with Oracle Integration*
Oracle CX Sales and B2B Service Adapter	*Using the Oracle CX Sales and B2B Service Adapter with Oracle Integration*
Oracle Database Adapter	*Using the Oracle Database Adapter with Oracle Integration*
Oracle Database Cloud Service Adapter	*Using the Oracle Database Cloud Service Adapter with Oracle Integration*
Oracle E-Business Suite Adapter	*Using the Oracle E-Business Suite Adapter with Oracle Integration*
Oracle Eloqua Cloud Adapter	*Using the Oracle Eloqua Cloud Adapter with Oracle Integration*
Oracle Enterprise Performance Management Cloud Adapter	*Using the Oracle Enterprise Performance Management Adapter with Oracle Integration*
Oracle ERP Cloud Adapter	*Using the Oracle ERP Cloud Adapter with Oracle Integration*
Oracle Field Service Cloud Adapter	*Using Oracle Field Service Cloud Adapter with Oracle Integration*
Oracle HCM Cloud Adapter	*Using the Oracle HCM Cloud Adapter with Oracle Integration*
Oracle Hospitality Adapter	*Using the Oracle Hospitality Adapter with Oracle Integration*
Oracle Intelligent Advisor Adapter	*Using the Oracle Intelligent Advisor Adapter with Oracle Integration*
Oracle Intelligent Track and Trace Adapter	*Using the Oracle Intelligent Track and Trace Adapter with Oracle Integration*
Oracle JD Edwards EnterpriseOne Adapter	*Using the Oracle JD Edwards EnterpriseOne Adapter with Oracle Integration*
Oracle Logistics Adapter	*Using the Oracle Logistics Adapter with Oracle Integration*
Oracle NetSuite Adapter	*Using the Oracle NetSuite Adapter with Oracle Integration*
Oracle Responsys Adapter	*Using the Oracle Responsys Adapter with Oracle Integration*
Oracle Service Cloud (RightNow) Adapter	*Using the Oracle Service Cloud (RightNow) Adapter with Oracle Integration*
Oracle Siebel Adapter	*Using the Oracle Siebel Adapter with Oracle Integration*
Oracle SOA Suite Adapter	*Using the Oracle SOA Suite Adapter with Oracle Integration*
Oracle Taleo Enterprise Edition Adapter	*Using the Oracle Taleo Enterprise Edition Adapter with Oracle Integration*

Adapter	For Information
Oracle Taleo Business Edition (TBE) Adapter	*Using the Oracle Taleo Business Edition (TBE) Adapter with Oracle Integration*
Oracle Unity Adapter	*Using the Oracle Unity Adapter with Oracle Integration*
Oracle Utilities Adapter	*Using the Oracle Utilities Adapter with Oracle Integration*
Oracle WebLogic JMS Adapter	*Using the Oracle WebLogic JMS Adapter with Oracle Integration*
PayPal Adapter	*Using the PayPal Adapter with Oracle Integration*
QuickBooks Adapter	*Using the QuickBooks Adapter with Oracle Integration*
REST Adapter	*Using the REST Adapter with Oracle Integration*
Salesforce Adapter	*Using the Salesforce Adapter with Oracle Integration*
Salesforce Commerce Cloud Adapter	*Using the Salesforce Commerce Cloud Adapter with Oracle Integration*
SAP Adapter	*Using the SAP Adapter with Oracle Integration*
SAP Ariba Adapter	*Using the SAP Ariba Adapter with Oracle Integration*
SAP ASE (Sybase) Adapter	*Using the SAP ASE (Sybase) Adapter with Oracle Integration*
SAP Commerce Cloud (Hybris) Adapter	*Using the SAP Commerce Cloud (Hybris)Adapter with Oracle Integration*
SAP Concur Adapter	*Using the SAP Concur Adapter with Oracle Integration*
SAP SuccessFactors Adapter	*Using the SAP SuccessFactors Adapter with Oracle Integration*
SAP S/4HANA Cloud Adapter	*Using the SAP S/4HANA Adapter with Oracle Integration*
ServiceNow Adapter	*Using the ServiceNow Adapter with Oracle Integration*
Shopify Adapter	*Using the Shopify Adapter with Oracle Integration*
Slack Adapter	N/A
SOAP Adapter	*Using the SOAP Adapter with Oracle Integration*
SugarCRM Adapter	*Using the SugarCRM Adapter with Oracle Integration*
SurveyMonkey Adapter	*Using the SurveyMonkey Adapter with Oracle Integration*
Trello Adapter	*Using the Trello Adapter with Oracle Integration*
Twilio Adapter	*Using the Twilio Adapter with Oracle Integration*
Twitter Adapter	*Using the Twitter Adapter with Oracle Integration*
UiPath Robotic Process Automation Adapter	*Using the UiPath Robotic Process Adapter with Oracle Integration*
Workday Adapter	*Using the Workday Adapter with Oracle Integration*
Zendesk Adapter	*Using the Zendesk Adapter with Oracle Integration*

You are now ready to configure connection security.

Configure Connection Security

Configure security for your connection by selecting the security policy and specifying the login credentials for the adapter you selected.

1. Go to the **Security** section.
2. See the following sections for information about selecting the security policy and entering your login credentials.

Adapter	For Information
Adobe Sign Adapter	*Using the Adobe Sign Adapter with Oracle Integration*
Apache Kafka Adapter	*Using the Apache Kafka Adapter with Oracle Integration*
AS2 Adapter	*Using the AS2 Adapter with Oracle Integration*
Automation Anywhere Adapter	*Using the Automation Anywhere Adapter with Oracle Integration*
Box Adapter	*Using the Box Adapter with Oracle Integration*
DocuSign Adapter	*Using the DocuSign Adapter with Oracle Integration*
eBay Marketplace Adapter	*Using the eBay Marketplace Adapter with Oracle Integration*
Eventbrite Adapter	*Using the Eventbrite Adapter with Oracle Integration*
Evernote Adapter	*Using the Evernote Adapter with Oracle Integration*
Facebook Adapter	*Using the Facebook Adapter with Oracle Integration*
File Adapter	NA
FTP Adapter	*Using the FTP Adapter with Oracle Integration*
Google Gmail Adapter	*Using the Google Gmail Adapter with Oracle Integration*
Google Calendar Adapter	*Using the Google Calendar Adapter with Oracle Integration*
Google Tasks Adapter	*Using the Google Tasks Adapter with Oracle Integration*
HubSpot Adapter	*Using the HubSpot Adapter with Oracle Integration*
Jira Adapter	*Using the Jira Adapter with Oracle Integration*
IBM DB2 Adapter	*Using the IBM DB2 Adapter with Oracle Integration*
IBM MQ Series JMS Adapter	*Using the IBM MQ Series JMS Adapter with Oracle Integration*
LinkedIn Adapter	*Using the LinkedIn Adapter with Oracle Integration*
Magento Commerce Cloud Adapter	*Using the Magento Commerce Cloud Adapter with Oracle Integration*
Mailchimp Adapter	*Using the Mailchimp Adapter with Oracle Integration*
Marketo Adapter	*Using the Marketo Adapter with Oracle Integration*

Adapter	For Information
Microsoft Office 365 Calendar Adapter	*Using the Microsoft Office 365 Calendar Adapter with Oracle Integration*
Microsoft Office 365 People Adapter	*Using the Microsoft Office 365 People Adapter with Oracle Integration*
Microsoft Office 365 Outlook Adapter	*Using the Microsoft Office 365 Outlook Adapter with Oracle Integration*
Microsoft SQL Server Adapter	*Using the Microsoft SQL Server Adapter with Oracle Integration*
MySQL Adapter	*Using the MySQL Adapter with Oracle Integration*
Oracle Advanced Queuing (AQ) Adapter	*Using the Oracle Advanced Queuing (AQ) Adapter with Oracle Integration*
Oracle Autonomous Data Warehouse Adapter	*Using the Oracle Autonomous Data Warehouse Adapter with Oracle Integration*
Oracle Autonomous Transaction Processing Adapter	*Using the Oracle Autonomous Transaction Processing Adapter with Oracle Integration*
Oracle Cloud Infrastructure Streaming Service Adapter	*Using the Oracle Cloud Infrastructure Streaming Service Adapter with Oracle Integration*
Oracle Commerce Cloud Adapter	*Using the Oracle Commerce Cloud Adapter with Oracle Integration*
Oracle CPQ Adapter	*Using the Oracle CPQ Adapter with Oracle Integration*
Oracle CX Sales and B2B Service Adapter	*Using the Oracle CX Sales and B2B Service Adapter with Oracle Integration*
Oracle Database Adapter	*Using the Oracle Database Adapter with Oracle Integration*
Oracle Database Cloud Service Adapter	*Using the Oracle Database Cloud Service Adapter with Oracle Integration*
Oracle E-Business Suite Adapter	*Using Oracle E-Business Suite Adapter with Oracle Integration*
Oracle Eloqua Cloud Adapter	*Using the Oracle Eloqua Cloud Adapter with Oracle Integration*
Oracle Enterprise Performance Management Cloud Adapter	*Using the Oracle Enterprise Performance Management Adapter with Oracle Integration*
Oracle ERP Cloud Adapter	*Using the Oracle ERP Cloud Adapter with Oracle Integration*
Oracle Field Service Cloud Adapter	*Using Oracle Field Service Cloud Adapter with Oracle Integration*
Oracle HCM Cloud Adapter	*Using the Oracle HCM Cloud Adapter with Oracle Integration*
Oracle Hospitality Adapter	*Using the Oracle Hospitality Adapter with Oracle Integration*
Oracle Intelligent Advisor Adapter	*Using the Oracle Intelligent Advisor Adapter with Oracle Integration*
Oracle Intelligent Track and Trace Adapter	*Using the Oracle Intelligent Track and Trace Adapter with Oracle Integration*
Oracle JD Edwards EnterpriseOne Adapter	*Using the JD Edwards EnterpriseOne Adapter with Oracle Integration*
Oracle Logistics Adapter	Using the Oracle Logistics Adapter with Oracle Integration
Oracle NetSuite Adapter	*Using the Oracle NetSuite Adapter with Oracle Integration*

Adapter	For Information
Oracle Responsys Adapter	*Using the Oracle Responsys Adapter with Oracle Integration*
Oracle Service Cloud (RightNow) Adapter	*Using the Oracle Service Cloud (RightNow) Adapter with Oracle Integration*
Oracle Siebel Adapter	*Using the Oracle Siebel Adapter with Oracle Integration*
Oracle SOA Suite Adapter	*Using the Oracle SOA Suite Adapter with Oracle Integration*
Oracle Taleo Enterprise Edition Adapter	*Using the Oracle Taleo Enterprise Edition Adapter with Oracle Integration*
Oracle Taleo Business Edition (TBE) Adapter	*Using the Oracle Taleo Business Edition (TBE) Adapter with Oracle Integration*
Oracle Unity Adapter	*Using the Oracle Unity Adapter with Oracle Integration*
Oracle Utilities Adapter	*Using the Oracle Utilities Adapter with Oracle Integration*
Oracle WebLogic JMS Adapter	*Using the Oracle WebLogic JMS Adapter with Oracle Integration*
PayPal Adapter	*Using the PayPal Adapter with Oracle Integration*
QuickBooks Adapter	*Using the QuickBooks Adapter with Oracle Integration*
REST Adapter	*Using the REST Adapter with Oracle Integration*
Salesforce Adapter	*Using the Salesforce Adapter with Oracle Integration*
Salesforce Commerce Cloud Adapter	*Using the Salesforce Commerce Cloud Adapter with Oracle Integration*
SAP Adapter	*Using the SAP Adapter with Oracle Integration*
SAP Ariba Adapter	*Using the SAP Ariba Adapter with Oracle Integration*
SAP ASE (Sybase) Adapter	*Using the SAP ASE (Sybase) Adapter with Oracle Integration*
SAP Commerce Cloud (Hybris) Adapter	*Using the SAP Commerce Cloud (Hybris) Adapter with Oracle Integration*
SAP Concur Adapter	*Using the SAP Concur Adapter with Oracle Integration*
SAP SuccessFactors Adapter	*Using the SAP SuccessFactors Adapter with Oracle Integration*
SAP S/4HANA Cloud Adapter	*Using the SAP S/4HANA Adapter with Oracle Integration*
Shopify Adapter	*Using the Shopify Adapter with Oracle Integration*
Slack Adapter	*Using the Slack Adapter with Oracle Integration*
SOAP Adapter	*Using the SOAP Adapter with Oracle Integration*
ServiceNow Adapter	*Using the ServiceNow Adapter with Oracle Integration*
SugarCRM Adapter	*Using the SugarCRM Adapter with Oracle Integration*
SurveyMonkey Adapter	*Using the SurveyMonkey Adapter with Oracle Integration*
Trello Adapter	*Using the Trello Adapter with Oracle Integration*
Twilio Adapter	*Using the Twilio Adapter with Oracle Integration*
Twitter Adapter	*Using the Twitter Adapter with Oracle Integration*

Adapter	For Information
UiPath Robotic Process Automation Adapter	*Using the UiPath Robotic Process Automation Adapter with Oracle Integration*
Workday Adapter	*Using the Workday Adapter with Oracle Integration*
Zendesk Adapter	*Using the Zendesk Adapter with Oracle Integration*

3. Click **Save**.

   For most adapters, you are now ready to test your connection. For some adapters, you must first configure an agent group.

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: • **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**, then click
   
   .

Edit a Connection

You can edit connection settings after creating a new connection.

1. In the left navigation pane, click **Home** > **Integrations** > **Connections**.
2. Go to the row of the connection to edit.
3. Click
   
   or click the connection name.

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

The Connection page is displayed.

4. Make any necessary edits.

   If you edit a connection currently used by an active integration, a dialog is displayed indicating that you must re-activate the integration for the connection updates to take effect.

Clone a Connection

You can clone a copy of an existing connection, even if the connection is locked. This provides a quick way to create a new connection.

1. In the navigation pane, click **Home** > **Integrations** > **Connections**.
2. Go to the row of the connection to clone.
3. Select **Clone** from the

   menu.

   The Clone Connection dialog is displayed.
4. Enter the connection information.
5. Click **Clone**.
6. Click

   to configure the credentials of your cloned connection. Cloning a connection does not copy the credentials.

See Editing a Connection for instructions.

Delete a Connection

You can delete a connection from the connection menu.

1. In the left navigation pane, click **Home** > **Integrations** > **Connections**.
2. Go to the row of the connection to delete.
3. Click **Delete** from the

   menu.

   The Delete Connection dialog is displayed if the connection is not used in an integration.
4. Click **Delete** to confirm deletion.

Unlock a Connection

When a connection is in edit mode and the browser crashes, the connection becomes locked, which prevents it from being edited.

To unlock the connection:

1. Log in again as the same user who was editing the connection when the browser crashed, then log out.

   For example, if you were logged in as icsdeveloper when the browser crashed, log in again as icsdeveloper, and log out. This action unlocks the connection.

or

1. Log in as a user with the Administrators role.
2. In the left navigation pane, click **Home** > **Integrations** > **Connections**.
3. Go to the row for the connection that is locked.
4. Select **Unlock** from the 
   menu.
5. Click **Unlock** to confirm the unlocking.

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.
   ```

Manage Security Certificates

You can manage security certificates in Oracle Integration.

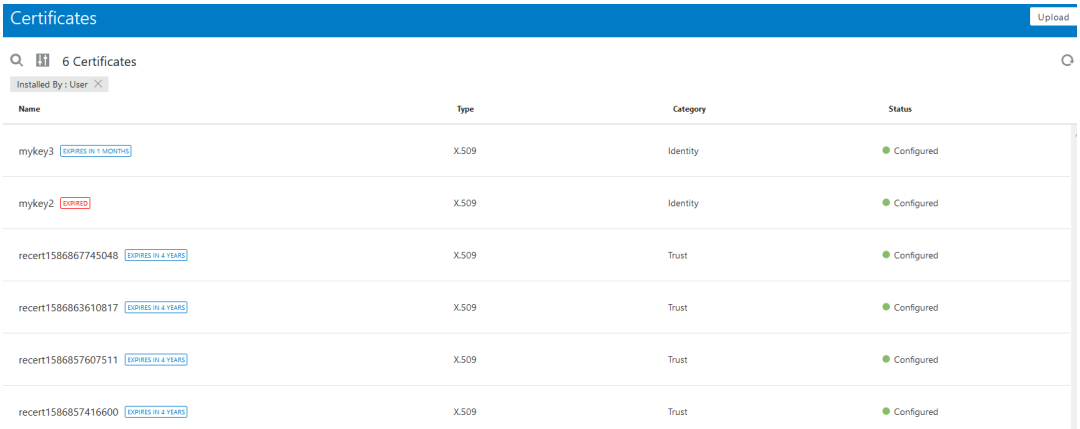
Topics:

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.



2. Click **Upload** at the top of the page.
   The Upload Certificate dialog box is displayed.
3. Enter an alias name and optional description.
4. 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)

1. Select a certificate category.
   a. **Trust**: Use this option to upload a trust certificate.
      i. Click **Browse**, then select the trust file (for example, `.cer` or `.crt`) to upload.
   b. **Identity**: Use this option to upload a certificate for two-way SSL communication.
      i. Click **Browse**, then select the keystore file (`.jks`) to upload.
      ii. 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.

      iii. Enter the password of the keystore being imported.
   c. Click **Upload**.

SAML (Authentication & Authorization)

1. 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.
   See Configure a 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**.

Update or Delete an SSL Certificate

You can update or delete certificates you uploaded into Oracle Integration. You cannot update or delete system certificates automatically included in Oracle Integration.

To update or delete a certificate:

1. Click **Home** > **Settings** > **Certificates**.
2. Identify the certificate you want to update or delete through either of the following methods:
   a. Scroll through the complete list or filter the display of system-provided or user-uploaded certifications by selecting > **Installed By**.
   b. Search by entering a partial or complete certificate name in the field. To remove search or filter criteria, click the **x** icon below the field.
3. At the far right of the certificate name, click the menu.
4. To update the certificate, click **Update**.
   - Update the certificate as required, such as uploading a new certificate. See Upload an SSL Certificate.

   > **Note:**
   > You cannot update an identity certificate. To change an identity certificate, you must first delete and then reupload it.

5. To delete the certificate, click **Delete**.
   - Click **Yes** when prompted to confirm your selection.

Configure Trigger Oracle Integration Messaging

Configure trigger Oracle Integration Messaging for your integration. This dialog is displayed when you add Oracle Integration Messaging as a trigger to an integration.

You can configure the trigger Oracle Integration Messaging. This enables you to subscribe to messages from Oracle Integration. Message subscription is accomplished through use of Oracle Integration Messaging.

Select the published integration to which to subscribe.

Element	Description
Select a Publisher	Select the published integration to which to subscribe. You must have already created a publisher to which to subscribe. The publisher does not need to be active, but must already be completely configured. After selecting a published integration, you perform the following tasks: • Add an invoke adapter to the integration to subscribe to the published integration. • Perform source-to-target mapping between the published integration and the invoke adapter. • Optionally configure source enrichments between the published integration and the invoke adapter.

See Oracle Integration Messaging, Create an Integration to Publish Messages to Oracle Integration, and Create an Integration to Subscribe to Oracle Integration.

3

Understand Integration Creation and Best Practices

Integrations use the connections you created to your applications, and define how information is shared between those applications. You can create new integrations, import integrations, modify or delete integrations, create integrations to publish messages, create integrations to subscribe to messages, and much more. Click one of the following topics to learn more about integration styles, best practices for designing an integration correctly from the beginning, trigger (inbound) and invoke (outbound) connections to your applications, and integration creation steps.

Topics:

Understand Integration Styles

You can select from several styles when creating an integration in the Select Integration Style dialog.

Style	Description
App Driven Orchestration	Create an integration that uses an event or a business object to trigger the integration. For example, you create an integration with an Oracle RightNow Adapter as a trigger and an Oracle CX Sales and B2B Service Adapter as an invoke. The Oracle RightNow Adapter subscribes to an event from the Oracle RightNow application to trigger the integration. Orchestration integrations include features such as the following: • Switch activities to create multiple routing expressions. • For-each activities for looping over repeating elements. • Assign activities for assigning values to scalar variables. • Ad-hoc mappings on switch branches. • Callback activities (to end a process and respond back to the sender) and end activities (to end a process without responding back to the sender) in asynchronous integrations. See Create Application-Driven Orchestrated Integrations.

Style	Description
Scheduled Orchestration	Create an integration that uses a schedule to trigger the integration instead of an adapter. For example, you add an initial invoke adapter to read a trigger file and a second FTP adapter to download the file for further processing. After designing this integration, you schedule when to run it. See Create Scheduled Integrations.
File Transfer	Create an integration to move files across a network. See Create Scheduled Integrations.
Basic Routing	Create an integration with a blank trigger and invoke in which to add your own adapters. You can also create a single routing expression and request and response enrichments, as needed. You cannot create multiple routing expressions. If your integration requires this feature, create an orchestrated integration. **Note**: Basic routing integrations in which you include the FTP Adapter to transfer files (sometimes known informally as file-A) have been deprecated. As an alternative, create an orchestrated, scheduled integration in which you include the FTP Adapter and a parallel-for-each action to transfer and manage files. Note that you can still clone and import this type of basic routing integration. See Create a Basic Routing Integration.
Publish to OIC	Create an integration in which you add a trigger adapter to publish messages to Oracle Integration through a predefined Oracle Integration Messaging invoke. No configuration of the invoke subscriber is required. The publisher and subscribers participating in this integration style can be activated and deactivated independently of each other. See Create an Integration to Publish Messages to Oracle Integration.
Subscribe to OIC	Create an integration in which you add an invoke adapter to subscribe to messages from Oracle Integration through an Oracle Integration Messaging trigger. You are prompted to select the publisher to which to subscribe. You must have already created a publisher to which to subscribe. The publisher does not need to be active, but must already be completely configured. Any business identifiers defined on fields in the published integration are copied to the subscriber. Any changes made to the published integration's business identifiers *after* copying are not reflected in the subscriber. The publisher and subscribers participating in this integration style can be activated and deactivated independently of each other. See Create an Integration to Subscribe to Oracle Integration.

Video

Common Integration Style Pitfalls and Design Best Practices

Note the following best practices and integration style pitfalls to avoid when designing an integration.

- Avoid Common Integration Style Pitfalls
- Avoid Creating Too Many Scheduled Integrations
- Synchronous Integration Best Practices
- Design Long-Running or Time-Consuming Integrations as Asynchronous Flows
- Time Outs in Service Calls During Synchronous Invocations
- Parallel Processing in Outbound Integrations

Avoid Common Integration Style Pitfalls

Designing integrations correctly from the start can save you a tremendous amount of rework. This section describes common integration style pitfalls (known as antipatterns) and best practices for avoiding these pitfalls.

- Chatty Integrations
- Scheduled Job that Never Stops Trying to Process
- Import an Externally Updated IAR File
- Synchronous Integration Doing Too Much
- Too Many Connections in an Integration
- Read Files with Many Records'
- Integrations Running Unchanged Despite Changing Business Needs

Chatty Integrations

Use Case: Synchronize records in a file or large data set with an external system (for example, synchronizing journal transactions or uploading employee records into Oracle HCM Cloud).

Antipattern	Why an Antipattern?	Best Practice
Use an invoke activity within a looping construct to call external APIs for every record.	• Downstream applications are receiving a large number of atomic requests. This puts the entire system under duress. • A usage-based pricing model translates to high costs.	• Leverage application capabilities to accept multiple records in a single request: – Salesforce: 200 records, Oracle Engagement Cloud/ERP Cloud: 100 records, Service Cloud: 1000 records • Leverage adapter capabilities to send a large data set as attachments/files. – Salesforce Adapter: 10,000 records/10 MB files, Oracle ERP Cloud Adapter supports FBDI files • Use a stage file action (the append to file option) - send the file to the destination at the end.

Scheduled Job that Never Stops Trying to Process

Use Case: Process records within a set of files with a tight service level agreement (SLA) (for example, synchronize employee records into Active Directory from Oracle HCM Cloud or send benefits information).

Antipattern	Why an Antipattern?	Best Practice
The scheduled integration looks for all files to process and loops over all to sequentially process until no files remain.	• If a large number of files exist, one run of a scheduled job executes for a long time and starves other jobs and may get terminated by the framework. • Processing tied to a single server: this does not leverage multiple nodes in a cluster.	• Limit the number of files to process in a single scheduled run. – Limit results from the list file. • Use schedule parameters to remember the last processed file for the next run. • Invoke the run now command from the same integration to trigger processing of the next file if waiting for the next scheduled run is not feasible.

Import an Externally Updated IAR File

Use Case: Need to leverage advanced XSL constructs that may not be available in the mapper.

Antipattern	Why an Antipattern?	Best Practice
Updating the IAR file externally and then importing it into Oracle Integration.	• This can lead to metadata inconsistency and validation failures. • Activation failures may occur.	• Use the import map feature in Oracle Integration. – Keeps metadata consistent and leverages validation.

Synchronous Integration Doing Too Much

Use Case: A request triggers complex processing involving enrichment and updates across multiple systems.

Antipattern	Why an Antipattern?	Best Practice
• Huge synchronous integrations modeling a large number of invokes / conditional logic. • Synchronous integrations with invokes within a loop with a large number of iterations.	• Susceptible to timeouts - Any marginal slowdown adds up. • Blocking call – Holds resources and starves other integrations.	• Explore moving completely to an asynchronous integration - fire and forget, asynchronous response. – Integration platform provides an acknowledgment to clients upon receiving the message. – Platform ensures guaranteed processing and also supports resubmission of failures. • Split into a synchronous integration containing mandatory processing before sending out a response and triggering separate asynchronous fire and forget integrations for other processing logic. • Optimize synchronous processing with a coarse-grained external API to replace multiple chatty calls.

Too Many Connections in an Integration

Use Case: As developers create integrations, they define their own connections pointing to the same application. This leads to many duplicate connections.

Antipattern	Why an Antipattern?	Best Practice
Every developer creates their own connection using a different set of configurations/credentials.	• High number of connections make manageability painful, especially when you need to update the endpoint, credentials, configuration, and so on. • Complicates impact analysis when there is an application upgrade or metadata/ coordinate change.	• Have a custodian create needed connections and ensure duplicate connections of the same types are not created. – Build a best practice for naming conventions and maintaining a set of configurations.

Read Files with Many Records

Use Case: Read a file with a large number of records and process individual records.

Antipattern	Why an Antipattern?	Best Practice
Reading the entire file in memory using the read file option and processing record by record.	• Consumes large amounts of memory and impacts other system processing. • Sequential processing does not leverage built-in map reduction capabilities.	• Download the file to the stage location using the download file option. – Stage files are secured and deleted at the completion of the instance. • Use the read file with segments option. – You can still model record level processing inside the segment processing. – Platform automatically processes segments in parallel. – Platform brings in only portions of the file to memory, as needed.

Integrations Running Unchanged Despite Changing Business Needs

Use Case: Integrations/schedules created during the initial implementation continue to run even though your business requirements have changed over time.

Antipattern	Why an Antipattern?	Best Practice
• Integrations and scheduled jobs created during the initial product implementation are never re-evaluated against changing business needs. • A proliferation of integrations occurs without looking at existing integrations.	• Unnecessary runs of jobs that handle no work. • Unoptimized invocations of multiple integrations with similar functionality. • Clutter with dead integrations, life cycle management (LCM) overheads, and developer confusion.	• Periodically analyze existing integrations/schedules against current business needs. – Look at monitoring data for historical runs. • Consolidate integrations that are very similar. – Integrations A and B subscribing to the Customer Created event in Oracle Engagement Cloud – integration A for invoking endpoint A and integration B for invoking endpoint B. • Deactivate integrations that are no longer relevant. – An integration for generating a file meant for partner consumption becomes irrelevant when the partner no longer needs it. • Adjust the schedules for integrations that need lesser frequency and delete schedules no longer required. – Files generated monthly have schedules run every few minutes. – Changes in business needs can make certain schedules completely unnecessary.

Avoid Creating Too Many Scheduled Integrations

When there are too many scheduled integrations configured, instances can get backlogged waiting for resources to become available or previous integration runs to complete. This can cause processing delays where some instances are in a waiting state longer than they should be and schedules may not start at the scheduled time.

As a best practice, it is recommended that you do not create too many scheduled integrations that are scheduled to run at the same time. Where possible:

- Use an asynchronous REST Adapter trigger instead of a scheduled trigger when an active schedule is not absolutely required.
- Do not create any long-running scheduled integrations (a scheduled integration that takes longer than one hour, for example, to complete). This blocks scheduler resources impacting other scheduled runs.
- Spread schedules out over time to avoid schedule clusters.

You can convert a scheduled integration to an application-driven orchestrated, REST Adapter trigger-based integration. See Convert a Scheduled Integration to a REST Adapter-Triggered Orchestration Integration.

If you absolutely require a large number of scheduled integrations, and you are encountering the issue described above, we recommend the following design changes as a solution:

1. For each scheduled integration, convert it to an application-driven orchestrated, REST Adapter trigger-based integration. See Convert a Scheduled Integration to a REST Adapter-Triggered Orchestration Integration.
2. Create a new scheduled integration that only performs an asynchronous invoke of the application-driven orchestrated integration you converted in step 1 above.
   This solution enables the scheduled integration to start at the scheduled time, invoke the REST Adapter trigger-based child integration asynchronously, and complete within milliseconds. This approach reduces the backlog and contention surrounding scheduler resources.

   If you have a large number of scheduled integrations to convert, a staged approach starting with the following integrations is recommended.

   - Longest-running scheduled integrations.
   - Scheduled integrations configured with the shortest frequency (for example, integrations that run every 10 minutes or less).

   Design any new scheduled integrations with the design practices described above.

See Scheduled Integrations Are Not Executing on Time and When a Scheduled Integration Instance Gets Terminated.

Synchronous Integration Best Practices

Note the following best practices when designing a synchronous integration.

- A synchronous integration calling any asynchronous request response service:
  - Calling asynchronous fire and forget (one-way) is acceptable.
  - Oracle Integration does not currently allow modeling an asynchronous request response service. However, all scheduled orchestration styles internally use an asynchronous request response. Therefore, a synchronous integration using a scheduled orchestration is an antipattern.
- A synchronous integration calling multiple services that span more than 5 minutes is reported as a stuck thread in Oracle WebLogic Server.

Design Long-Running or Time-Consuming Integrations as Asynchronous Flows

Note the following best practice when designing long-running or time-consuming integrations.

Do not expose long-running or time-consuming integrations as synchronous flows. This action can lead to client applications (including other integrations) timing out. Synchronous integrations also have a server-side time out. Instead, model any synchronous integration taking more than two minutes as an asynchronous flow.

Time Outs in Service Calls During Synchronous Invocations

You may have scenarios in which synchronous invocations from Oracle Integration (including calls to other integrations) are blocking calls and must complete within 300 seconds.

Because the call may involve one or more proxies, each of the proxies may have a similar time out. For instance, the default proxy on Oracle Public Cloud has a time out value of 120 seconds. If the call is to an on-premises service behind a fire wall, the proxy configured may also have its own time out value.

In case time outs are defined at multiple layers, the service invocation fails at the first time out.

Parallel Processing in Outbound Integrations

While there is no specific integration design to automatically enable an outbound integration to send data to different third party systems in parallel, there are integration design approaches that allow this scenario.

Separate the integration into multiple integrations:

- Create a main parent integration that only receives/processes the data.
- Create separate child integrations to perform the individual outbound REST invocations.

The interface between the main and separate child integrations can follow these approaches:

- Consist of dummy REST calls, but it must be asynchronous. Essentially, the asynchronous calls are not blocked by the response and the fire-and-forget design enables available threads to work on child integration processing in parallel, within the available system resources. This type of design is recommended because if all synchronous REST calls are done in the same integration, a time out error may occur if the sum of time taken for each synchronous call exceeds five minutes.
- Follow a publish/subscribe design approach (for example, putting the data events in a queue, having each child flow subscribe from the queue, and so on).

Understand Trigger and Invoke Connections

When you design an integration, you drag trigger (source) and invoke (target) adapter connections into the integration. The information required to connect to the application is already defined in the connection. However, you still must specify certain information, such business object, business event, operation, or other elements to use for the request and response and how to process the data. This invokes the Adapter Endpoint Configuration Wizard that guides you through configuration of the adapter connections.

For example, you design an integration that synchronizes customer information between an Oracle Service Cloud (RightNow) application (the trigger) and an Oracle CX Sales and B2B Service application (the invoke) in real time. Whenever an Organization is created in Oracle Service Cloud (RightNow), an AccountService is created in Oracle CX Sales and B2B Service in real time. To achieve this scenario, you define both an Oracle Service Cloud (RightNow) Adapter as a trigger connection and an Oracle CX Sales and B2B Service Adapter as an invoke connection in the Adapter Endpoint Configuration Wizard.

Trigger Connections

You can define the following adapter connections as triggers in an integration. The trigger (source) connection sends requests to Oracle Integration. The following guides describe the pages of the Adapter Endpoint Configuration Wizard.

Adapter	For Information
AS2 Adapter	*Using the AS2 Adapter with Oracle Integration*
Box Adapter	*Using the Box Adapter with Oracle Integration*
File Adapter	*Using the File Adapter with Oracle Integration*
IBM DB2 Adapter	*Using the IBM DB2 Adapter with Oracle Integration*
IBM MQ Series JMS Adapter	*Using the IBM MQ Series JMS Adapter with Oracle Integration*
Microsoft SQL Server Adapter	*Using the Microsoft SQL Server Adapter with Oracle Integration*
MySQL Adapter	*Using the MySQL Adapter with Oracle Integration*
Oracle Advanced Queuing (AQ) Adapter	*Using the Oracle Advanced Queuing (AQ) Adapter with Oracle Integration*
Oracle Cloud Infrastructure Streaming Service Adapter	*Using the Oracle Cloud Infrastructure Streaming Service Adapter with Oracle Integration*
Oracle Commerce Cloud Adapter	*Using the Oracle Commerce Cloud Adapter with Oracle Integration*
Oracle CPQ Adapter	*Using the Oracle CPQ Adapter with Oracle Integration*
Oracle CX Sales and B2B Service Adapter	*Using the Oracle CX Sales and B2B Service Adapter with Oracle Integration*
Oracle Database Adapter	*Using the Oracle Database Adapter with Oracle Integration*
Oracle E-Business Suite Adapter	*Using Oracle E-Business Suite Adapter with Oracle Integration*
Oracle Eloqua Cloud Adapter	*Using the Oracle Eloqua Cloud Adapter with Oracle Integration*
Oracle ERP Cloud Adapter	*Using the Oracle ERP Cloud Adapter with Oracle Integration*
Oracle HCM Cloud Adapter	*Using the Oracle HCM Cloud Adapter with Oracle Integration*
Oracle Hospitality Adapter	*Using the Oracle Hospitality Adapter with Oracle Integration*
Oracle Intelligent Advisor Adapter	*Using the Oracle Intelligent Advisor Adapter with Oracle Integration*
Oracle JD Edwards EnterpriseOne Adapter	*Using the JD Edwards EnterpriseOne Adapter with Oracle Integration*
Oracle Logistics Adapter	*Oracle Logistics Adapter with Oracle Integration*
Oracle Service Cloud (RightNow) Adapter	*Using the Oracle Service Cloud (RightNow) Adapter with Oracle Integration*
Oracle Siebel Adapter	*Using the Oracle Siebel Adapter with Oracle Integration*
Oracle WebLogic JMS Adapter	*Using the Oracle WebLogic JMS Adapter with Oracle Integration*
Oracle Utilities Adapter	*Using the Oracle Utilities Adapter with Oracle Integration*
REST Adapter	*Using the REST Adapter with Oracle Integration*

Adapter	For Information
Salesforce Adapter	*Using the Salesforce Adapter with Oracle Integration*
SAP Adapter	*Using the SAP Adapter with Oracle Integration*
SAP ASE (Sybase) Adapter	*Using the SAP ASE (Sybase) Adapter with Oracle Integration*
ServiceNow Adapter	*Using the ServiceNow Adapter with Oracle Integration*
Slack Adapter	*Using the Slack Adapter with Oracle Integration*
SOAP Adapter	*Using the SOAP Adapter with Oracle Integration*

Invoke Connections

You can define the following adapter connections as invokes in an integration. Oracle Integration sends requests or information to the invoke (target) connection. The following guides describe the pages of the Adapter Endpoint Configuration Wizard.

Adapter	For Information
Adobe Sign Adapter	*Using the Adobe Sign Adapter with Oracle Integration*
Apache Kafka Adapter	*Using the Apache Kafka Adapter with Oracle Integration*
AS2 Adapter	*Using the AS2 Adapter with Oracle Integration*
Automation Anywhere Adapter	*Using the Automation Anywhere Adapter with Oracle Integration*
Box Adapter	*Using the Box Adapter with Oracle Integration*
DocuSign Adapter	*Using the DocuSign Adapter with Oracle Integration*
eBay Marketplace Adapter	*Using the eBay Marketplace Adapter with Oracle Integration*
Eventbrite Adapter	*Using the Eventbrite Adapter with Oracle Integration*
Evernote Adapter	*Using the Evernote Adapter with Oracle Integration*
Facebook Adapter	*Using the Facebook Adapter with Oracle Integration*
File Adapter	*Using the File Adapter with Oracle Integration*
FTP Adapter	*Using the FTP Adapter with Oracle Integration* (Basic Routing Integration) *Using the FTP Adapter with Oracle Integration* (App Driven Integration)
Google Gmail Adapter	*Using the Google Gmail Adapter with Oracle Integration*
Google Calendar Adapter	*Using the Google Calendar Adapter with Oracle Integration*
Google Tasks Adapter	*Using the Google Tasks Adapter with Oracle Integration*
HubSpot Adapter	*Using the HubSpot Adapter with Oracle Integration*
Jira Adapter	*Using the Jira Adapter with Oracle Integration*
IBM DB2 Adapter	*Using the IBM DB2 Adapter with Oracle Integration*
IBM MQ Series JMS Adapter	*Using the IBM MQ Series JMS Adapter with Oracle Integration*

Adapter	For Information
LinkedIn Adapter	*Using the LinkedIn Adapter with Oracle Integration*
Mailchimp Adapter	*Using the Mailchimp Adapter with Oracle Integration*
Marketo Adapter	*Using the Marketo Adapter with Oracle Integration*
Microsoft Office 365 Calendar Adapter	*Using the Microsoft Office 365 Calendar Adapter with Oracle Integration*
Microsoft Office 365 People Adapter	*Using the Microsoft Office 365 People Adapter with Oracle Integration*
Microsoft Office 365 Outlook Adapter	*Using the Microsoft Office 365 Outlook Adapter with Oracle Integration*
Microsoft SQL Server Adapter	*Using the Microsoft SQL Server Adapter with Oracle Integration*
MySQL Adapter	*Using the MySQL Adapter with Oracle Integration*
Oracle Advanced Queuing (AQ) Adapter	*Using the Oracle Advanced Queuing (AQ) Adapter with Oracle Integration*
Oracle Autonomous Data Warehouse Adapter	*Using the Oracle Autonomous Data Warehouse Adapter with Oracle Integration*
Oracle Autonomous Transaction Processing Adapter	*Using the Oracle Autonomous Transaction Processing Adapter with Oracle Integration*
Oracle Cloud Infrastructure Streaming Service Adapter	*Using the Oracle Cloud Infrastructure Streaming Service Adapter with Oracle Integration*
Oracle Commerce Cloud Adapter	*Using the Oracle Commerce Cloud Adapter with Oracle Integration*
Oracle CPQ Adapter	*Using the Oracle CPQ Adapter with Oracle Integration*
Oracle CX Sales and B2B Service Adapter	*Using the Oracle CX Sales and B2B Service Adapter with Oracle Integration*
Oracle Database Adapter	*Using the Oracle Database Adapter with Oracle Integration*
Oracle Eloqua Cloud Adapter	*Using the Oracle Eloqua Cloud Adapter with Oracle Integration*
Oracle E-Business Suite Adapter	*Using Oracle E-Business Suite Adapter with Oracle Integration*
Oracle ERP Cloud Adapter	*Using the Oracle ERP Cloud Adapter with Oracle Integration*
Oracle Field Service Cloud Adapter	*Using the Oracle Field Service Cloud Adapter with Oracle Integration*
Oracle HCM Cloud Adapter	*Using the Oracle HCM Cloud Adapter with Oracle Integration*
Oracle Hospitality Adapter	*Using the Oracle Hospitality Adapter with Oracle Integration*
Oracle Intelligent Advisor Adapter	*Using the Oracle Intelligent Advisor Adapter with Oracle Integration*
Oracle Intelligent Track and Trace Adapter	*Using the Oracle Intelligent Track and Trace Adapter with Oracle Integration*
Oracle JD Edwards EnterpriseOne Adapter	*Using the JD Edwards EnterpriseOne Adapter with Oracle Integration*
Oracle Logistics Adapter	*Using the Oracle Logistics Cloud Adapter with Oracle Integration*
Oracle NetSuite Adapter	*Using the Oracle NetSuite Adapter with Oracle Integration*

Adapter	For Information
Oracle Responsys Adapter	*Using the Oracle Responsys Adapter with Oracle Integration*
Oracle Service Cloud (RightNow) Adapter	*Using the Oracle Service Cloud (RightNow) Adapter with Oracle Integration*
Oracle Siebel Adapter	*Using the Oracle Siebel Adapter with Oracle Integration*
Oracle SOA Suite Adapter	*Using the Oracle SOA Suite Adapter with Oracle Integration*
Oracle Taleo Enterprise Edition Adapter	*Using the Oracle Talent Acquisition Cloud (Taleo EE) Adapter with Oracle Integration*
Oracle Taleo Business Edition (TBE) Adapter	*Using the Oracle Taleo Business Edition (TBE) Adapter with Oracle Integration*
Oracle Unity Adapter	*Using the Oracle Unity Adapter with Oracle Integration*
Oracle Utilities Adapter	*Using the Oracle Utilities Adapter with Oracle Integration*
Oracle WebLogic JMS Adapter	*Using the Oracle WebLogic JMS Adapter with Oracle Integration*
PayPal Adapter	*Using the PayPal Adapter with Oracle Integration*
QuickBooks Adapter	*Using the QuickBooks Adapter with Oracle Integration*
REST Adapter	*Using the REST Adapter with Oracle Integration*
Salesforce Adapter	*Using the Salesforce Adapter with Oracle Integration*
Salesforce Commerce Cloud Adapter	*Using the Salesforce Commerce Cloud Adapter with Oracle Integration*
SAP Adapter	*Using the SAP Adapter with Oracle Integration*
SAP Ariba Adapter	*Using the SAP Ariba Adapter with Oracle Integration*
SAP ASE (Sybase) Adapter	*Using the SAP ASE (Sybase) Adapter with Oracle Integration*
SAP Commerce Cloud (Hybris) Adapter	*Using the SAP Commerce Cloud (Hybris) Adapter with Oracle Integration*
SAP Concur Adapter	*Using the SAP Concur Adapter with Oracle Integration*
SAP SuccessFactors Adapter	*Using the SAP SuccessFactors Adapter with Oracle Integration*
SAP S/4HANA Cloud Adapter	*Using the S/4HANA Adapter with Oracle Integration*
ServiceNow Adapter	*Using the ServiceNow Adapter with Oracle Integration*
Shopify Adapter	*Using the Shopify Adapter with Oracle Integration*
Slack Adapter	*Using the Slack Adapter with Oracle Integration*
SOAP Adapter	*Using the SOAP Adapter with Oracle Integration*
SugarCRM Adapter	*Using the SugarCRM Adapter with Oracle Integration*
SurveyMonkey Adapter	*Using the SurveyMonkey Adapter with Oracle Integration*
Trello Adapter	*Using the Trello Adapter with Oracle Integration*
Twilio Adapter	*Using the Twilio Adapter with Oracle Integration*
Twitter Adapter	*Using the Twitter Adapter with Oracle Integration*

Adapter	For Information
UiPath Robotic Process Automation Adapter	*Using the UiPath Robotic Process Automation Adapter with Oracle Integration*
Workday Adapter	*Using the Workday Adapter with Oracle Integration*
Zendesk Adapter	*Using the Zendesk Adapter with Oracle Integration*

Create Integrations

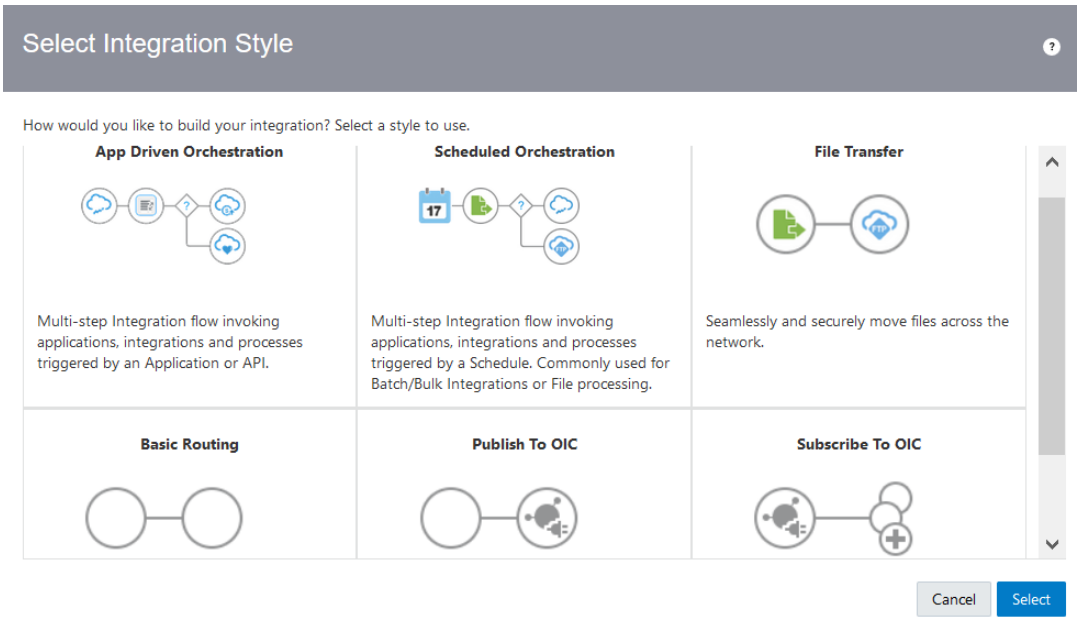
Creating an integration includes defining the trigger and invoke application connections, and defining how data is mapped between the two applications. The procedure below provides general instructions for creating an integration, with links to more detailed information for certain steps. As you perform each step, the progress indicator changes to let you know how close you are to completing the integration.

If you want to use a lookup table in your data mapping, create the lookup first. See Creating Lookups for instructions.

To create an integration:

1. In the left navigation pane, click **Home** > **Integrations** > **Integrations**.
2. On the Integrations page, click **Create**.

   The Select Integration Style dialog is displayed.



3. Select the type of integration style applicable to your business needs, and click **Select**. See Understand Integration Styles.

> **Note:**
>
> The Basic Routing integration style has been deprecated. Oracle recommends that you use the App Driven Orchestration integration style, which provides more flexibility.

The Create New Integration dialog is displayed.

4. Enter the following information:

Field	Description
What do you want to call your integration?	Provide a meaningful name so that others can understand the integration. You can include English alphabetic characters, numbers, underscores, and dashes in the identifier.
Identifier	Accept the default identifier value. The identifier is the same as the integration name you provided, but in upper case.
Version	Accept the default version number of `01.00.0000`. Or, if you want to change the version number, enter the version using numbers only in this format: `xx.xx.xxxx`. Integrations are uniquely identified by an identifier and version. Note the version format of `xx.yy.zzzz`, where `xx` is the major version and `yy.zzzz` is the minor version. Integrations having the same identifier, but a different major version, can be active at the same time. For example, `INT-A/1.00.0000` and `INT-A/2.00.0000` can be active at the same time. When activating an integration while another integration of the same identifier and same major version is already active, the currently activated integration is deactivated prior to activating the selected integration. For example, if two integrations have the following integration states: • `INT-A/2.00.0000` - Not active • `INT-A/2.10.0000` - Not active Integration `INT-A/2.00.0000` is then activated. • `INT-A/2.00.0000` is now active. • `INT-A/2.10.0000` is not active. Integration `2.10.0000` is then activated. • `INT-A/2.00.0000` is now not active. • `INT-A/2.10.0000` is now active.
Documentation URL	Optionally enter a URL to a design document that provides the details of the integration. The valid URL format is: `http(s)://`*`hostname`*`:`*`port`*`/`*`documentation_path`*

Field	Description
What does this integration do?	Provide a meaningful description so that others can understand the integration.
Which package does this integration belong to?	Enter a new or existing package name in which to place your integration. As you enter the initial letters of an existing package, it is displayed for selection. See Manage Packages and About Oracle Integration Packages.
Which keyword defines this integration	Enter keywords (tags) to identify the integration. You can search for integrations on the Integrations page using keywords.

5. Click **Create**.
6. See the following sections based on your selection.

If You Selected...	See...
App Driven Orchestration	Create an Orchestrated Integration
Scheduled Orchestration	Create a Scheduled Integration
File Transfer	Create a Scheduled Integration
Basic Routing	Create Basic Routing Integrations
Publish to OIC	Create an Integration to Publish Messages to Oracle Integration
Subscribe to OIC	Create an Integration to Subscribe to Oracle Integration

4

Create Application-Driven Orchestrated Integrations

You can create business object- or event-based orchestrated integrations. Orchestrated integrations can be synchronous, asynchronous, or fire-and-forget types. Orchestrated integrations use Oracle BPEL Process Manager capabilities. Oracle BPEL Process Manager enables you to define how a business process that involves web services is executed. BPEL messages invoke remote services and orchestrate process execution. When designing integrations, you can create multiple routing expressions.

> **Note:**
>
> Ensure that you optimize integration design. If an orchestrated integration contains too many actions (for example, there are 25 or more stage write actions), activation failure can occur. For example, split an integration with many actions into multiple integrations. See Troubleshoot Integration Activations.

Topics:

Create an Orchestrated Integration

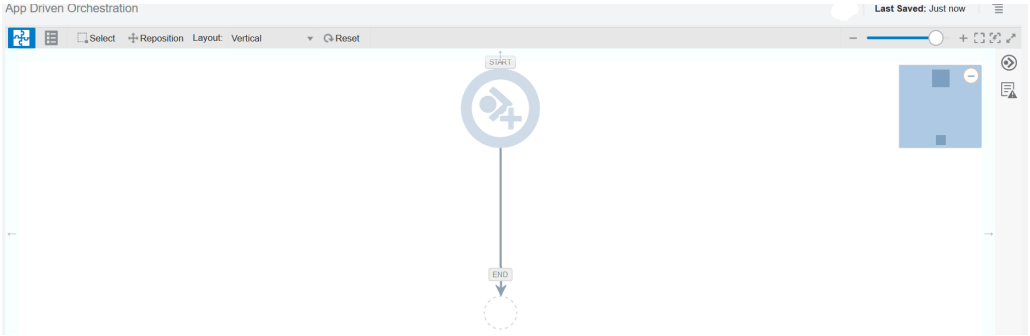
This section describes how to create an orchestrated integration. It also provides an overview of the orchestrated integration canvas.

To create an orchestrated integration:

1. Follow the steps in Create Integrations to create an orchestrated integration.

   An empty integration canvas with the following sections is displayed:

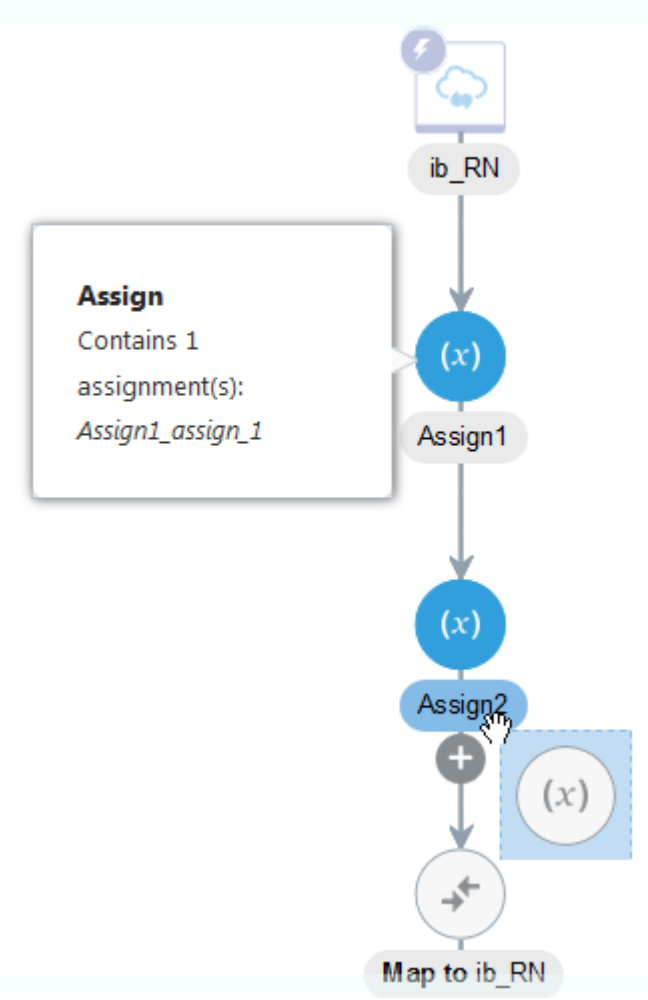
- The  icon in the upper right corner enables you to display a menu of available trigger adapter connections (for example, Oracle CX Sales and B2B Service Adapter). Click an adapter to display the number of configured adapter connections available for adding to an integration. A trigger enables you to create an inbound connection in an integration.
- The empty integration is identified by a **START** label in which you can drag the trigger to define the inbound part of the integration. You can also place your cursor over the **+** sign to invoke an inline menu for adding a trigger. See Add Actions, Connections, and Artifacts Through an Inline Menu.



- Several icons are provided in the upper left corner for working with the integration.
  - 
    Canvas View: Displays the default view of the integration canvas.
  - 
    Pseudo View: Displays the integration vertically with child nodes indented. Details about each node in the integration are displayed to the right. You can edit the integration by selecting the **+** sign to invoke an inline menu for adding invokes and actions. See Add Actions, Connections, and Artifacts Through an Inline Menu. When you select **Pseudo View**, the options for **Layout View** disappear.

  - **Global Fault**: Select to add a global fault to the integration.

- **Select**: Select to move specific portions of an integration.
- **Reposition**: Select to enter reposition mode, then select the action or invoke in the integration and drag it to a different part of the integration. You can reposition invokes and assign, function call, map, notification, stage file (except for the Read File in Segments operation), scope, and wait actions. You cannot reposition stop, return, or error hospital elements; and repeat elements such as for-each, while, and switch actions. For this example, **Assign1** is being moved below **Assign2**.



When the action or invoke is moved, the integration design is refreshed. When you click **Save**, validation errors or warnings caused by the move are displayed in the **Errors** palette and on the individual actions or invokes. The Errors palette shows the element names. The error/warning count is displayed at the top right. The count represents the number of actions that have errors/warnings. If you fix all errors, and only warnings still exist, the count of actions that have warnings is displayed. Click **Reposition** again to return to edit mode to resolve any errors.

- **Layout**: Select the option for viewing the integration layout:
  - **Horizontal**: Displays the integration horizontally.
  - **Vertical**: Displays the integration vertically.
- **Reset**: Click to reset the integration to its normal size and place it on the left side of the page.

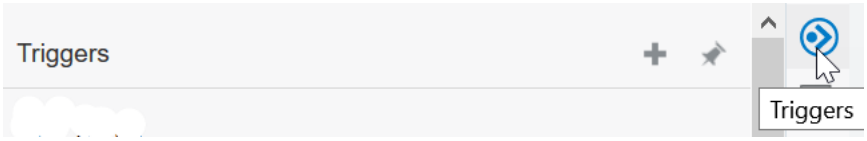
- Several icons are provided in the upper right corner of the integration for adjusting the size of the integration and getting started with integration design.
  - **Zoom Out**: Click to decrease the size of the integration.
  - **Sizing Bar**: Drag the circle up and down the bar to change the size of the integration.
  - **Zoom In**: Click to increase the size of the integration.
  - **Zoom To Fit**: Click to make the entire integration visible on the page.
  - **Go To Trigger**: Click to go to the trigger connection in the integration.
  - **Maximize / Minimize**: Click to maximize the size of the orchestration. Click again to minimize the size of the orchestration.
  - A box below these icons shows a scale model of the undefined integration. You can place you cursor within the box or anywhere in the canvas to move your integration. You can also drag parts of the integration (such as switch actions) around the canvas to redraw the integration. However, the order of the integration does not change.
  - **Triggers**: Click to display a list of trigger connections from which to choose.
  - **Errors**: Displays an errors in the integration.

Define Inbound Triggers and Outbound Invokes

To define inbound triggers and outbound invokes:

1. To add a trigger connection to the integration canvas, click the **+** sign below **START** in the integration canvas or click  on the right side of the canvas. Both options show a list of available trigger connections.

   If the adapter connection you want to use does not exist, click the **+** sign in the **Triggers** section to create a new connection.

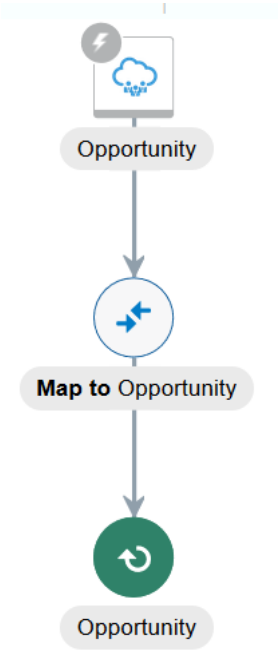
   The Create Connection — Select Adapter dialog is displayed for selecting the connection. Once the adapter connection is configured, tested, and the message indicates that it is 100% complete, save and close the page. The new adapter connection is displayed in the panel. An **Edit** icon is also displayed to the right of the connection name for updates. If you edit the connection so that it is now *not* 100% complete, then save and close the Connections page, a message is displayed in the banner indicating that it is not 100% complete and cannot be displayed in the panel. For the connection to be displayed in the panel again, you must return to the Connections page and configure the connection to be 100% complete.

2. Add the configured adapter to the large **+** section within the circle in the integration canvas.

   This invokes the Adapter Endpoint Configuration Wizard.

3. Complete the pages of the wizard to configure the selected adapter. For this example, an Oracle HCM Cloud Adapter is selected in which a request opportunity business object and a delayed response are configured. See Understand Trigger and Invoke Connections.

When complete, a configured trigger is displayed in the canvas. An unconfigured mapper icon is displayed in the middle. Because this trigger was configured to send a response, a return icon is displayed in green in the integration canvas. Green indicates that design is complete. You *cannot* delete a trigger in an integration (no delete option is available).



> **Note:**
>
> On the right side, the
> 
> triggers icon is replaced by the
> 
> invokes icon that enables you to add multiple outbound invoke connections to the integration.

An

icon is now displayed below

. When expanded, this section displays the following options:

- **Assign**: Enables you to assign variables to variables using the Expression Builder.
- **Callback**: Enables you to end a process and return to the trigger. For example, you can add a switch activity and define a branch in which you add a **Callback**. If some defined expression logic is not met, this branch is taken. The integration is stopped and the trigger receives a response indicating that the integration is not continuing.
- **B2B**: Translate a message to or from the Electronic Data Interchange (EDI) format in the integration. This action converts an EDI document into an Oracle Integration message, or vice versa.

- **For Each**: Enables you to loop over a repeating element and execute one or more actions within the scope of the for-each action. The number of looping iterations is based on a user-selected repeating element.
- **Integration**: Invokes another integration from an orchestration integration. When you drag this action into the integration, you are prompted to select an active integration to invoke.
- **Javascript**: Enables you to add JavaScript functions to the integration. This enables you to call JavaScript functions from an integration.
- **Logger**: Enables you to log messages to the activity stream and diagnostic logs by adding the logger action at any point in the integration. You can create a log message that is a static message or variable-populated message in the Expression Builder.
- **Map**: Adds mappings to the integration, as needed. Mappings are also automatically added when you add triggers, invokes, and some actions to an integration. In most cases, the messages you transfer between applications in an integration have different data structures. Mappings enable you to define how data is transferred, or mapped, between applications.
- **Note**: Adds placeholder notes similar to sticky notes to an integration. For example, you have not yet defined an invoke connection and want to add a placeholder note in the integration indicating that you plan to define the invoke connection later.
- **Notification**: Enables you to send a notification email to relevant users at specific points in the execution of an integration. You can set the To, From, and Subject parts of an email. You can also create the body part of an email using parameters defined in the Expression Builder.
- **Process**: Invokes a process from an orchestration integration. When you drag this action into the integration, you are prompted to select an application workspace and a process to invoke.
- **Return**: Enables you to return an immediate response.
- **Re-throw Fault**: Sends failed messages to the error hospital for further analysis. If the integration contains a defined global fault, the error captured by the re-throw fault action is sent through the global fault and onto the error hospital for analysis. If no global fault is defined, the fault is sent.
- **Scope**: Manages a group of actions. The scope action is a collection of child actions and invokes that can have their own fault handlers. The scope action provides the behavior context for the child elements. The elements defined in the parent scope have local visibility inside the scope.
- **Stage File**: Enables you to process files in integrations. The stage file action can process each file downloaded by the FTP Adapter. The stage file action can read (and remove any trailer), write, zip, and unzip files in a stage integration known to Oracle Integration.
- **Stitch**: Enables you to incrementally build a message payload from one or more existing payloads. The stitch action provides an editor that enables you to assign values to variables.
- **Switch**: Defines branches to add routing expression in the integration. You define one or more branches. An otherwise branch is taken if the routing expressions for the initial branches do not resolve to true.
- **Stop**: Enables you to terminate the integration. No response message is returned to the trigger.
- **Throw New Fault**: Generates an error and configures a code, reason, details, and skip condition for that error at a certain point during the execution of an integration.

- **Wait**: Enables you to delay the execution of an integration for a specified period of time. A typical use for this action is to invoke a specific operation at a certain time.
- **While**: Enables you to loop over actions or invoke connections as long as a specific condition is met. You define the condition for the while loop in the Expression Builder.

4. On the right side of the canvas, click 
to expand the panel. As with the trigger connection, you can click the **+** sign to create a new invoke connection and click the **Edit** icon to edit an existing connection. See Step 1 for information.
5. Click the adapter type to display the specific type and number of configured adapters.
6. Drag the specific configured adapter to the integration canvas. As you do, two large **+** sections within circles are displayed:
   - A section before the request mapper.
   - A section after the request mapper.
7. Drop the adapter in the appropriate section. For this example, the invoke is added before the request mapper.

   This invokes the Adapter Endpoint Configuration Wizard.
8. Complete the pages of the wizard to configure the selected adapter. For this example, an Oracle CX Sales and B2B Service Adapter named Order with a selected business object is defined for a synchronous response. See Understand Trigger and Invoke Connections.

   When complete, a configured invoke connection is displayed in the canvas.

   You can click the trigger and invoke connections to edit or view their contents. If you re-edit the selected connection in the Adapter Endpoint Configuration wizard and click **Done** (even without making any changes) you are prompted to update the configuration and regenerate the artifacts.
   - If you select **Yes** after making only minor changes, the system validates the flow and displays the warnings and errors (if there are any) so you can fix any potential problems.
   - If you select **Yes** after making major changes to the trigger connection (for example, changing the message exchange pattern of the trigger from synchronous to asynchronous), all mappings, replies, and stop elements are deleted from the orchestration except for the system-generated reply or stop at the end of the integration. Tracking information is also deleted.

     Editing an invoke connection can also result in major change. However, the impact is minimal compared to editing a trigger. Invoke editing does not have flow-wide consequences such as with maps and other nodes being removed.

   You can place your cursor anywhere in the canvas to move the integration as needed. You can also move the integration from within the large box in the upper right corner.

Receive Requests for Multiple Resources in a Single REST Adapter Trigger Connection

You can expose multiple entry points to a single orchestrated integration that uses the REST Adapter as the trigger connection. Each entry point can be configured with a different resource

URI and HTTP action/verb, as necessary. This feature eliminates the need to create multiple integrations (each with a separate resource URI and verb) to perform different operations.

This feature is useful in the following scenarios:

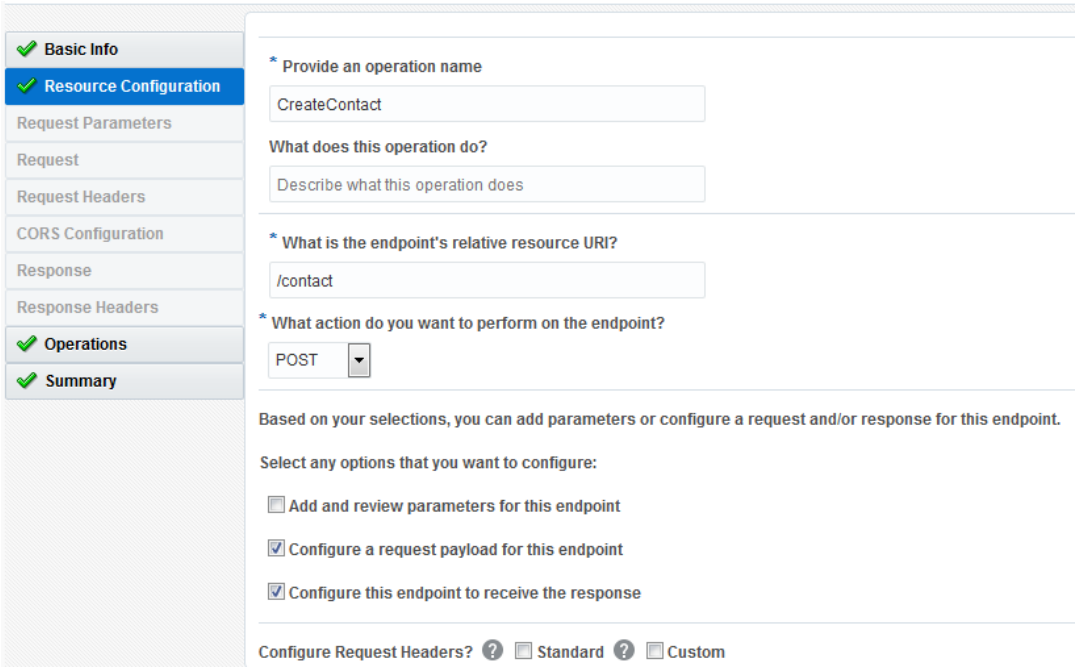
- Supports multiple resource URIs and actions/verbs in a specific integration through the REST Adapter.
- Supports multiple integration styles (for example, synchronous and asynchronous). The combination of a resource and a verb is called an operation. Each operation can be configured as a synchronous or an asynchronous one-way service. Asynchronous services accept the request and return immediately with an HTTP `202` status.

> **Note:**
>
> This feature is only available when creating the **App Driven Orchestration** integration style.

To configure this feature in an orchestrated integration:

1. Select to create an **App Driven Orchestration** integration style.
2. Drag a REST Adapter into the canvas as a trigger connection.

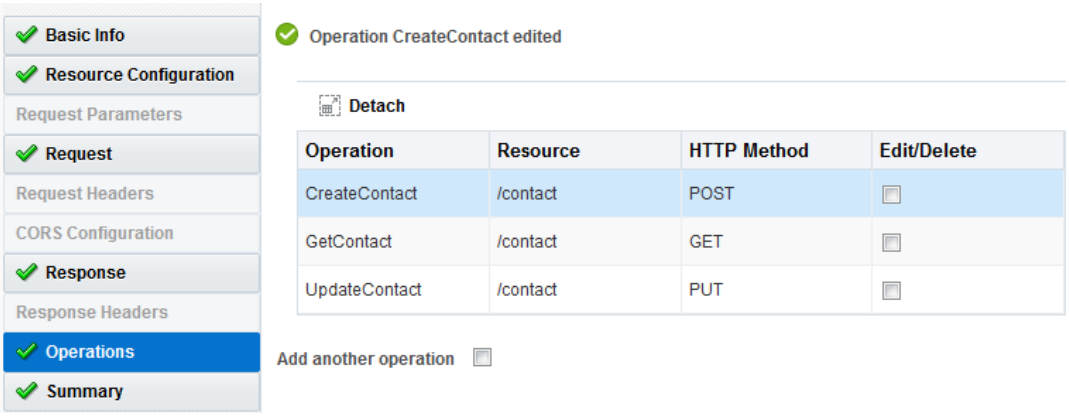
   This invokes the Adapter Endpoint Configuration Wizard.
3. On the Basic Info page, specify the following:
   a. Enter a name and optional description.
   b. Select the **Select to configure multiple resources or verbs** checkbox.
4. On the Resource Configuration page, specify the operation name, the endpoint relative resource URI, and the action/verb to perform, then click **Next**.



5. On the Operations page, click **Add another operation** to add another resource URI and action/verb, then click **Next**.

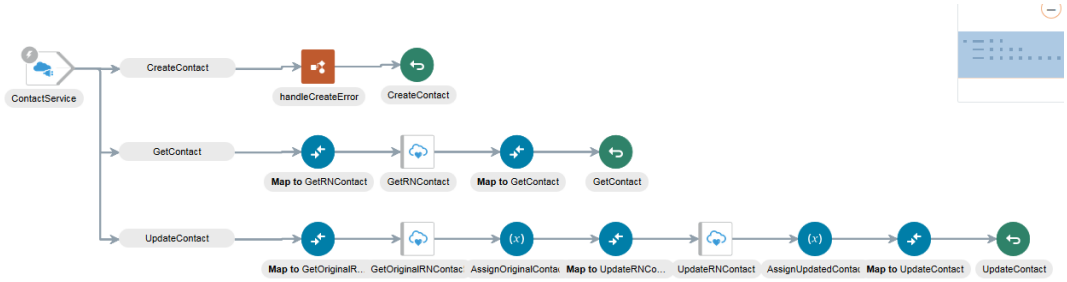
You are returned to the Resource Configuration page.

6. On the Resource Configuration page, specify a different operation name, endpoint relative resource URI, and action/verb to perform, then click **Next**.
7. Continue this process to add as many operations as necessary.
8. On the Operations page, note that all operations are displayed. Each operation represents a different branch in a single integration. The maximum number of operations (branches) you can create in one integration is eleven.

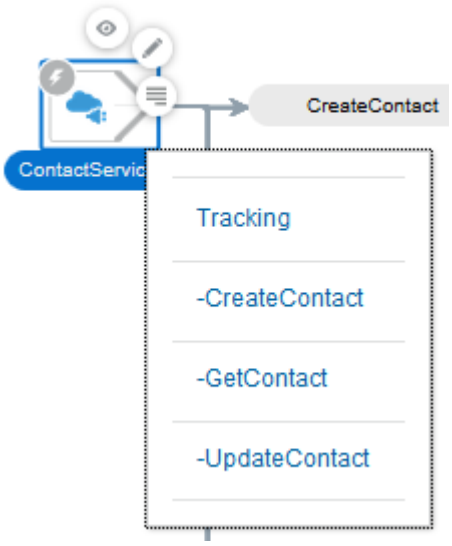


9. On the Summary page, review your selections, and click **Done**.

   For this example, there are three branches (**CreateContact**, **GetContact**, and **UpdateContact**).
10. Place your cursor over each branch to show details about the operation/resource name, action/verb, and resource URI.
11. Further configure your integration by dragging appropriate actions *after* each operation name. You cannot add actions between the REST Adapter icon and the operation name.



12. Highlight the REST Adapter, and click the **More Actions** option to show a menu that lists all operation names.
13. Click the operation name for which to define business identifier tracking variables.



14. Set the tracking variables for each operation branch. Tracking variables are populated as per the operation. You must set primary tracking variable for all operations to activate the integration. If any branches are not configured with tracking variables, a red error icon is displayed on the node. Place your cursor over the error icon to show a detailed error message.
15. Activate and invoke the integration to take the appropriate branch.

    Once activated, the adapter metadata page displays information corresponding to each of the configured resources and verbs.
    The swagger displays all the configured resources and the verbs for each such resource. Just like swagger from a single resource REST trigger endpoint, you can use this swagger in another invoke connection to automatically discover and browse the resource implemented.

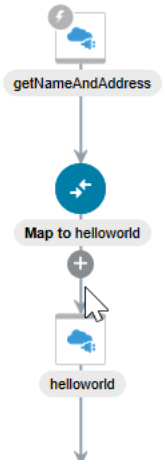
> **Note:**
>
> You can edit the action in the integration to remove or add operation branches. This enables you to convert to and from a standard trigger-based integration with a single operation to an integration with multiple operation branches.

Add Actions, Connections, and Artifacts Through an Inline Menu

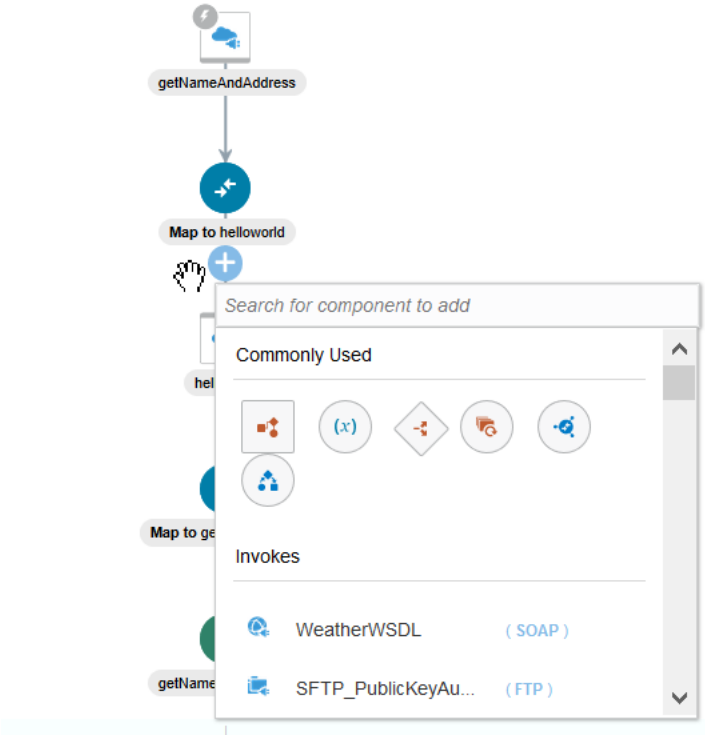
As an alternative to dragging actions, trigger and invoke connections, and integration artifacts from the right navigation pane, you can hover your cursor over the **+** sign that is displayed between the nodes in your integration. When you click the **+** sign, a list of actions, connections, and artifacts available for adding to your integration is displayed. A search facility is also available.

To add actions, connections, and artifacts through an inline menu:

1. Hover your cursor over the lines between the nodes to display a **+** sign.
2. Click the **+** sign.

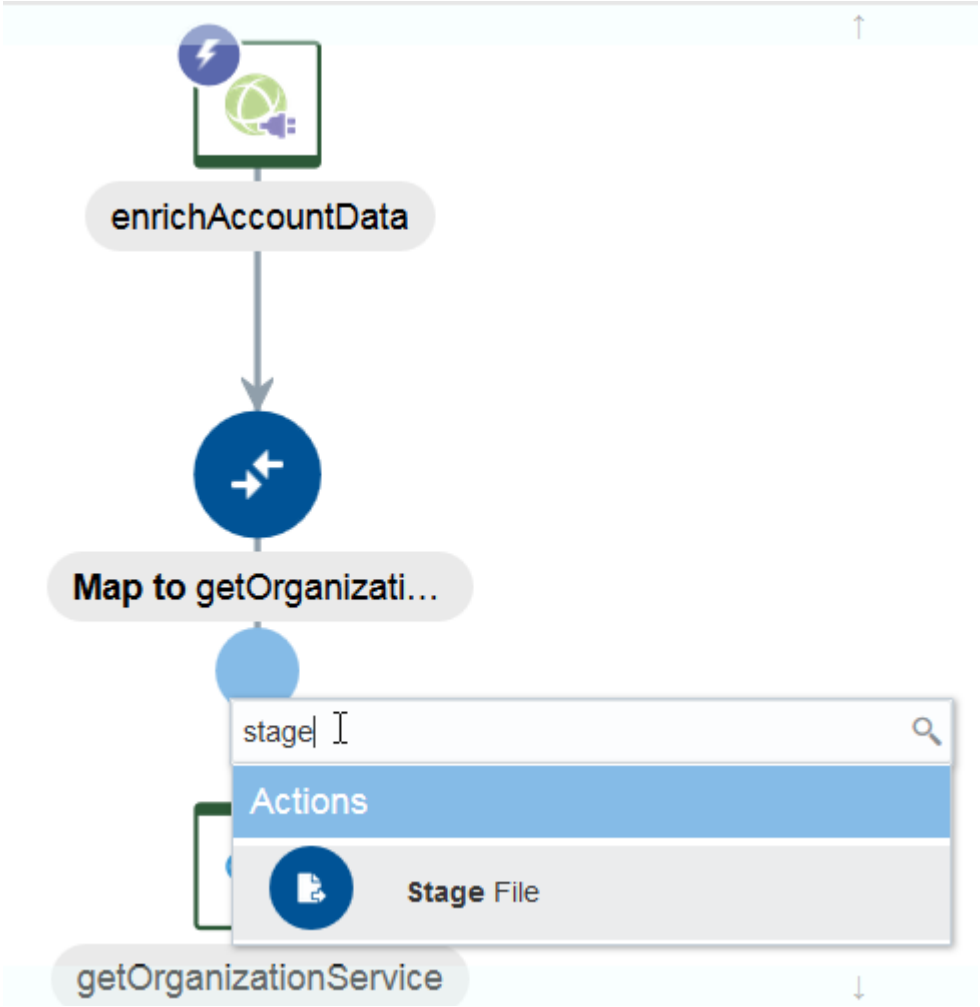


A menu with the following selections and a search field are displayed.



Element	Description
Commonly Used	Displays actions that are commonly placed between the two nodes.
Connections	Displays adapter connections available to add. The option is only displayed when a connection can be added after the nodes.
Actions	Displays actions that are allowed to be placed between the two nodes.

3. Select the appropriate element to add to your integration or begin entering the element type in the search field at the bottom to show matching elements to add.



The dialog box or wizard specific to your selection is displayed for configuration. For example, if you add a connection, the Adapter Endpoint Configuration Wizard is displayed.

Create Global Variables

You can create complex and simple type global variables that are available for usage throughout an orchestrated integration (for example, when building an expression in the Expression Builder of an assign action).

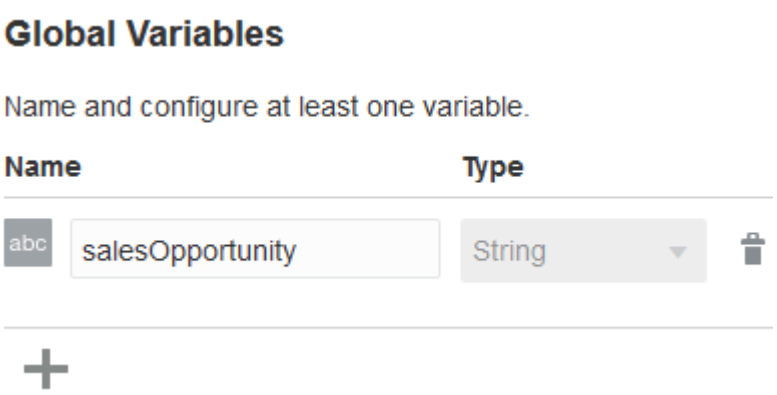
User-defined complex type variables are defined based on the WSDLs/XSDs exposed in the integration by the trigger connection and any invokes connection. These variables are then available throughout the integration for assignment and usage.
To create global variables:

1. In the upper right corner of the integration canvas, click .

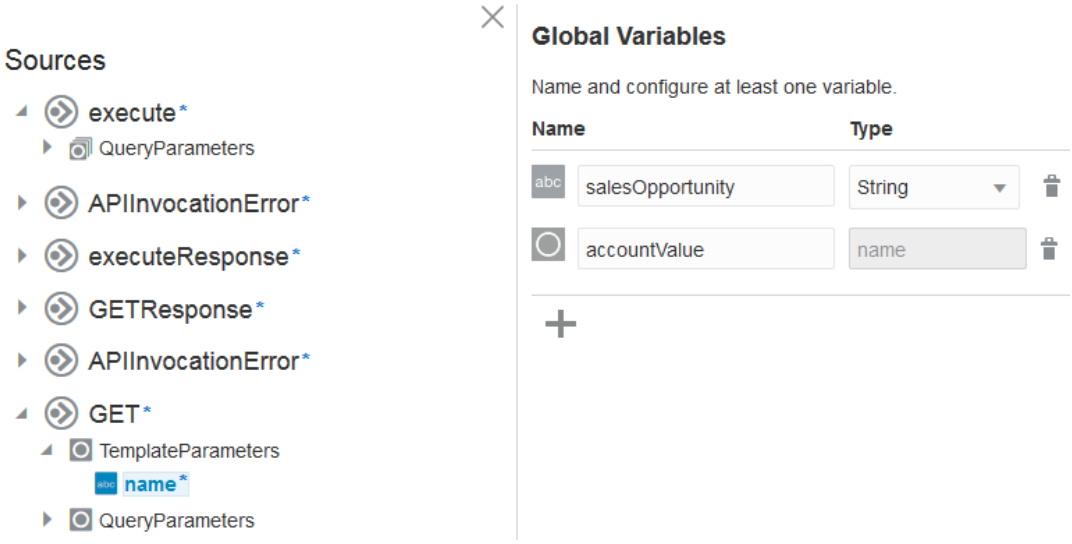
   If there are no global variables, the Global Variables dialog is displayed.

2. Click **Add Variable**.

3. Enter a name and select the data type from the dropdown list. The **Object** selection is considered a complex type.



4. Click **+** to add more global variables. A maximum of 20 global variables can be added.
5. If you add a global variable that is an **Object** data type, the dialog is expanded to display a source tree with all the elements from the request object.
6. Expand the tree and click any element (root/child). This marks the current global variable type as a selected node type. The **Type** list selection changes from **Object** to the node name that you select.



Note the following complex data type design rules:

- Complex types are only based on XML root elements. If you select a nonroot element when creating a complex type global variable, a WSDL is automatically generated defining it as a root element for use throughout the integration.
- Because complex type global variables are based on WSDLs/XSDs introduced into the integration by the trigger connection and invoke connections, there may be an impact when the trigger or an invoke is deleted, modified, or regenerated. Global variables react to these design changes, which may result in invalid usage/assignments.

7. Click to save your changes and close the dialog.

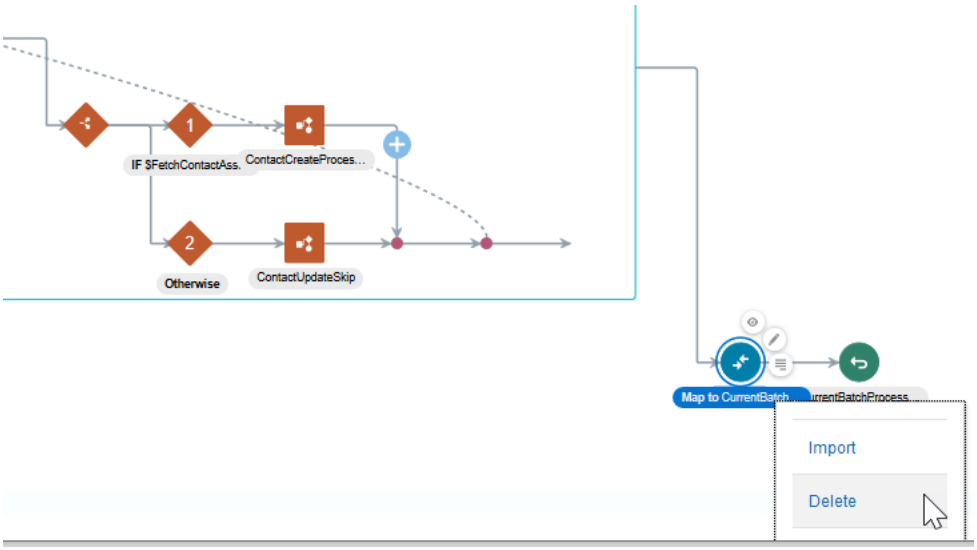
You cannot edit a variable. Instead, click to delete the variable, then recreate the variable.

Define Ad-Hoc Mappings

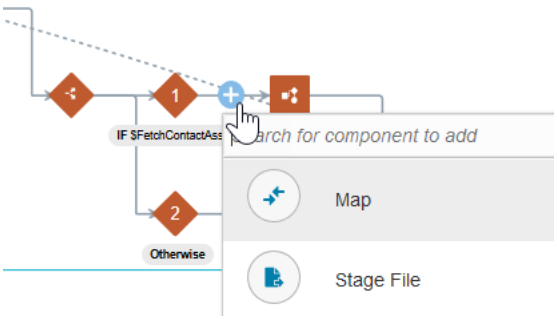
As you add switches and their associated invoke connections to the switch branches, you can add ad-hoc mappers, as needed. You can also delete the mappers that were automatically created when you added your first trigger to the integration, if they are not needed.

To define ad-hoc mappings:

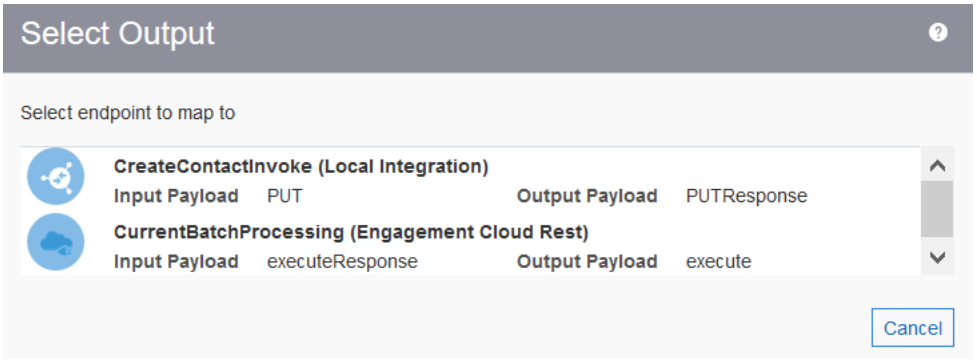
1. Delete any mappings that are not needed. For this example, the following mapping is deleted.



2. Add a map action to an integration in either of the following ways:
   - On the right side of the canvas, click
     and drag the **Map** action to the appropriate location.
   - Click at the location where you want to add the map action, then select **Map**.



The Select Output dialog is displayed.



3. Select the endpoint to which to map.

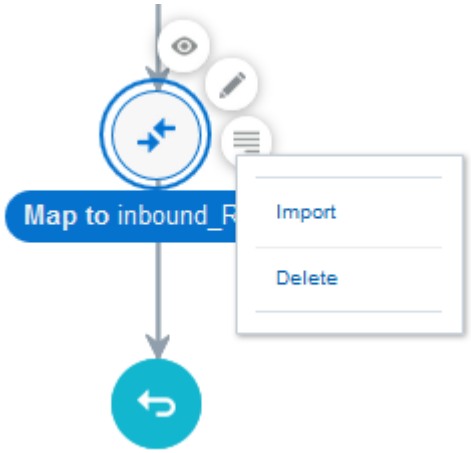
   The mapper is displayed.

4. Map appropriate elements from the source data structure to the target data structure.
5. When complete, click **Close**, then click **Apply** when prompted to save your changes.

Import a Map File into an Orchestrated Integration

You can import an XSL map file that was previously exported from the *same* integration. This action overwrites the existing mapping file. For example, you can export the map from a specific integration, edit the XSL file as per a user requirement, save it, and import it back into the same integration. You cannot import an XSL map file into an orchestrated integration that was exported from a different integration in Oracle Integration or from an application in Oracle JDeveloper.

1. Right-click the map in which you want to import an integration, and select **More Actions** > **Import**.



2. Browse for the map file to import, then click **Import**. You only import the map file of an exported integration into Oracle Integration. You do *not* import the entire integration in which the map file is included into Oracle Integration.

Add Actions to an Orchestrated Integration

You can add actions to your integrations.

Topics

Loop over Repeating Elements with a For-Each Action

The for-each action enables you to loop over a repeating element and execute one or more actions within the scope of the for-each action. The number of loop iterations is based on a user-selected repeating element. For example, you may have an integration in which you have downloaded a number of files and want to loop over the output of the files. The for-each action enables you to perform this task.

> **Note:**
>
> When you configure a stop action inside a for-each action, the entire integration is terminated when the for-each action is executed for the first time. The for-each action is not allowed to execute more than once. The stop action does not display any message describing this behavior during design time.

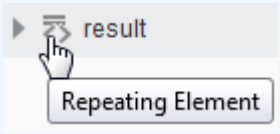
Creating a For-Each Action

1. Add a for-each action to an integration in either of the following ways:

- On the right side of the canvas, click 
  and drag the **For Each** action to the appropriate location.
- Click  at the location where you want to add the for-each action, then select **For Each**.

The For Each dialog is displayed.

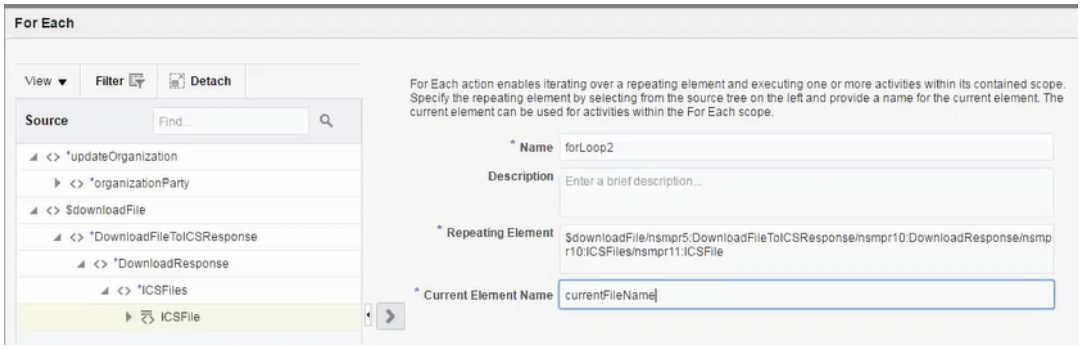
2. Expand the **Source** tree to select an element.
3. Drag a repeatable element to the **Repeating Element** field. This is the element over which to loop.

> **Note:**
>
> Note the following restrictions:
>
> - The selected element must be repetitive. You can identify repetitive elements by the two-bar icon to the left of the element name.
>
> 
>
>
> - Any parent of the selected element must not be repetitive.
> - The data type of the selected element must be scalar.
> - Global and nonglobal repeated elements can be selected.
> - If you have a repeating element within another repeating element (that is, a list within a list), you must first create a for-each action and loop over the parent list. This gives you access to the child list during every iteration. You can then create a second for-each action within the scope of the first for-each action and loop over the child list.

For this example, the element over which to loop is **ICSFile**.

4. Enter a name in the **Name** field and an optional description of the action in the **Description** field.

   For every iteration of the loop, there is a single reference to the repeating element (**ICSFile**). A current element name file is required for this action to occur.
5. Enter an alias for the current file of the iteration in the **Current Element Name** field.



Note:

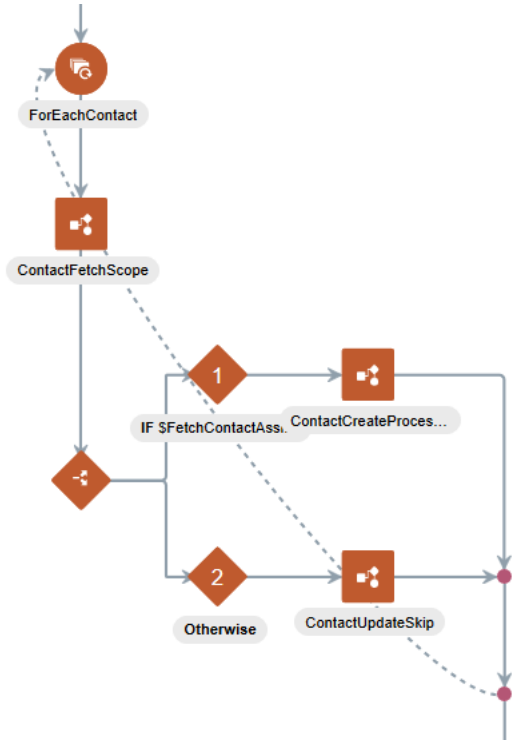
If you drag a for-each action into a scheduled integration or the for-each action is not inside a while action, for-each action, scope action, and so on, an additional field called **Process items in parallel** is visible at the bottom of the dialog.

When selected, iterations of the for-each action are run in parallel. For example, if an integration processes a list of files with a for-each loop created over every file, the processing of the entire file is done in parallel for every file. If a variable is declared outside a for-each loop with **Process items in parallel** selected and updated within the for-each loop, the last updated value for that variable is available outside of the loop. Also, the degree of parallelism (DOP) is set to `1` to avoid concurrency issues.

6. Click **Done**.

   The for-each action is displayed in the canvas. A looping arrow indicates that this action performs repetitive looping.

7. Drag other actions inside the for-each action to define what should happen during each iteration of the loop. For this example, scope and switch actions are defined within the for-each action.



Tracking the Status of a For-Each Action During Runtime

You can track the status of the for-each action in the Track Instances page through the tracking diagram and activity stream for an activated integration. This is only possible if there is a tracking instance.

1. In the left navigation pane, click **Home** > **Monitoring** > **Integrations** > **Tracking**.
2. Click the business identifier value of the integration to track.

   The integration flow (including any for-each actions) is displayed. Any for-each action failures are identified by red arrows.

3. Select **View Activity Stream** from the menu.

   Details about processing status (including any for-each actions) are displayed, including any failures. If failures are inside the for-each action, the iteration number and other details are displayed. If processing is successful, no details and counts are displayed.

Related Documentation

The following sections provide examples of using a for-each action in an integration:

- Use Bulk Response Operations in an Integration in *Using the Salesforce Adapter with Oracle Integration Generation 2*
- Process Large Data Sets Asynchronously with Different Bulk Import Operations in *Using the Marketo Adapter with Oracle Integration Generation 2*
- Subscribe to Atom Feeds in a Scheduled Integration in *Using the Oracle HCM Cloud Adapter with Oracle Integration Generation 2*
- Invoke an Endpoint Dynamically in *Using the Oracle ERP Cloud Adapter with Oracle Integration Generation 2*
- Create an Integration to Import and Process Bulk Files in *Using the FTP Adapter with Oracle Integration Generation 2*

Video

Route Expressions with Switch Action Branches

You can define switch action branches to add routing expressions in your integration. The switch action takes the first branch that evaluates to true. The other branches are ignored. A switch is essentially single-threaded.

To define switch action branches:

1. Add a switch action to an integration in either of the following ways:
   - On the right side of the canvas, click 
     and drag the **Switch** action to the appropriate location.
   - Click  at the location where you want to add the switch action, then select **Switch**.

   > **Note:**
   >
   > Nested switches are supported.

   Two branches are automatically created:
   - **Undefined** (first) branch: You must define a routing expression for this branch.
   - **Otherwise** (second) branch: This branch is taken if the routing expression for the initial branch does not resolve to true.

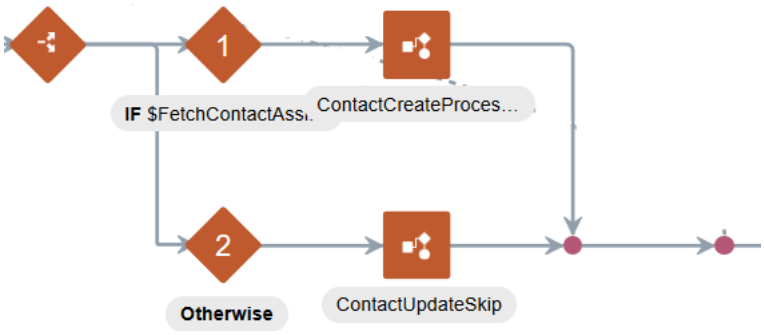
   > **Note:**
   >
   > To add more branches, click the question mark in the switch activity to invoke a menu.

2. Click the **Undefined** branch icon.
3. Select the **Edit** icon from the menu that is displayed. This invokes the Expression Builder.
4. Define a routing expression, then click **Close**. XPath version 2.0 functions are supported. Base 64 encode and decode functions that process data confidentially across layers and functions that return boolean results are also supported. Functions that return nonboolean values are not supported.

   For this example, the following expression is defined:

   ```
   $FetchContactAssign = "failed"
   ```

   You can now define different data flows for both the defined and otherwise branches in the switch activity.
5. Drag other actions and invokes into the switch action as required for your integration. For this example, scope actions are defined for both switch action branches.



Related Documentation
The following sections provide examples of using a switch action in an integration:

- Invoke an Endpoint Dynamically in *Using the Oracle ERP Cloud Adapter with Oracle Integration Generation 2*
- Design an Asynchronous Search Operation with Pagination in *Using the Oracle NetSuite Adapter with Oracle Integration Generation 2*
- Use the Extract API in a Scheduled Orchestrated Integration in *Using the SAP Concur Adapter with Oracle Integration Generation 2*
- Use the Pagination Concept in an Integration in *Using the Jira Adapter with Oracle Integration Generation 2*

Video

Manage a Group of Actions and Fault Handlers with a Scope Action

You can manage a group of actions with a scope action. The scope action is essentially a collection of child actions and invokes that can have their own fault handlers. The scope action provides the behavior context for the child elements. The elements defined in the parent scope have local visibility inside this scope. Anything that you can do in an integration such as designing invokes, mappings, and actions can be done in a scope action.

> **Note:**
>
> Scopes can have fault handlers in which specific faults can be caught and rethrown. However, in the case of connectivity agent-based invokes, the named fault handlers are not executed. All fault handling must be done in the default fault handler.

- Creating a Scope Action
- Add Nested Scopes to a Scope Action
- Reposition a Scope Action in an Integration
- Tracking the Status of a Scope Action During Runtime
- Renaming a Scope

Creating a Scope Action

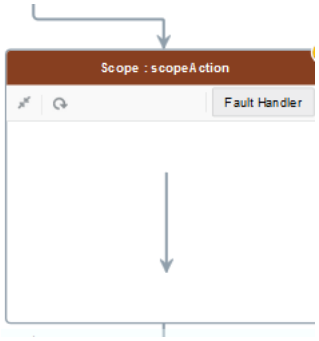
To create a scope action:

1. Add a scope action to an integration in either of the following ways:

- On the right side of the canvas, click 
  and drag the **Scope** action to the appropriate location.
- Click  at the location where you want to add the scope action, then select **Scope**.

2. Enter a name and optional description when prompted, and click **Create**.

   The action is displayed in the integration.



3. Drag and design appropriate actions, mappings, and invokes into the scope action.

   When you add invokes to a scope, the named faults associated with the invokes are added for selection to the **Fault Handler** section of the scope. These are faults that the invoke can handle. The uniqueness of the named faults is defined by the qname of the fault. If there are multiple invokes that define the qname fault, the fault (at runtime) can respond to any invoke. The various invokes cannot be differentiated if their qnames are not unique.

4. Select appropriate named faults.

   You can also define catch all blocks for fault handling through use of the raise error action in the **Fault Handler** section of the scope action. Catch all blocks are processed if an invoke throws a fault, but there is not a specific catch named for it. See Catch Faults with a Re-throw Fault Action.

5. Collapse the scope action by clicking the **collapse** icon in the upper left corner to condense the view of your integration. You can expand it again by clicking the **expand** icon.

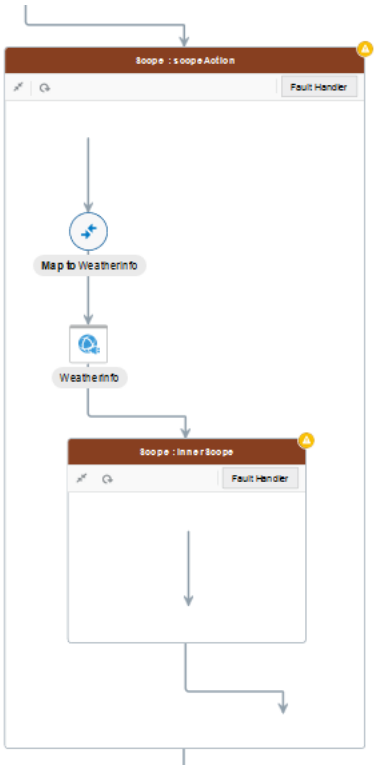
> **Note:**
>
> Note the following issues when using the REST Adapter with fault handling.
>
> - For orchestrated integrations with multiple REST Adapter invoke connections, only one handler can be defined when there are multiple invokes in an integration with scopes. Though the fault handler options are available for each invoke in the scope, it always points to one single handler.
> - If you add a new fault handler into a 17.4.5 integration using an old adapter configuration (for example, a pre-17.4.5 configuration from a prebuilt or older integration), the fault handler does not work. Before adding a fault handler to the REST Adapter endpoints, ensure that the endpoints were created/edited in version 17.4.5 or later.

Add Nested Scopes to a Scope Action

You can add nested (child) scope actions to a basic scope action. This provides a more sophisticated way of organizing or separating actions into a subsection of the integration. A nested scope provides the following capabilities:

- Behaves the same way as a basic scope. It provides its own container of child actions and fault handlers.
- There is no limitation to the levels of nesting. Even a scope's fault handlers can have nested scopes.

1. Drag a scope inside an existing scope.



2. Add actions and fault handlers to the nested scope, as necessary.

Reposition a Scope Action in an Integration

You can reposition a scope action in an integration. For example, you can move a scope from one part of an integration into a branch of a switch action.

1. Find the scope action to reposition.
2. Inside the scope, click
   
   . This collapses the scope.
3. Click **Reposition** above the integration canvas to enter reposition mode.
4. Drag the scope action to the **plus** sign location inside the switch action at which to place it.
5. After repositioning is complete, click **Reposition** to return to regular integration design mode.

> **Note:**
>
> Only collapsed scopes can currently be repositioned. Other container actions such as while actions, switch actions, for-each actions, and so on cannot be repositioned. You can move noncontainer actions (for example, assign actions, notification actions, and so on) by using the same reposition mechanism as with a scope action.

Tracking the Status of a Scope Action During Runtime

You can track the status of the scope action in the Tracking Details page through the tracking diagram and activity stream for an activated integration. This is only possible if there is a tracking instance.

1. In the left navigation pane, select **Home** > **Monitoring** > **Integrations** > **Tracking**.
2. Click the business identifier value of the integration to track.

   Depending upon runtime execution, several scope execution states can be displayed in the diagram:

   - Scope execution succeeds and is displayed in green. Because the fault handler is not executed, the **Fault Handler** section of the scope remains hidden.
   - Scope execution fails and is displayed in red, but the fault handler succeeds in handling the fault and is displayed in green. Execution continues after the scope. Because the fault handler was executed, the **Fault Handler** section of the scope is visible.
   - Both the scope and fault handler fail. Both are displayed in red. Both the scope and the **Fault Handler** section are displayed.

3. Select **View Activity Stream** from the menu.
   Details about processing status are displayed.

See Track Business Identifiers in Integrations During Runtime for additional details.

Renaming a Scope

1. Open the integration that includes the scope to rename.
2. Click
   to collapse the scope action.
3. Click the scope and select
   .
4. Change the name and click **Save**.

Assign Values to Scalar Variables in an Assign Action

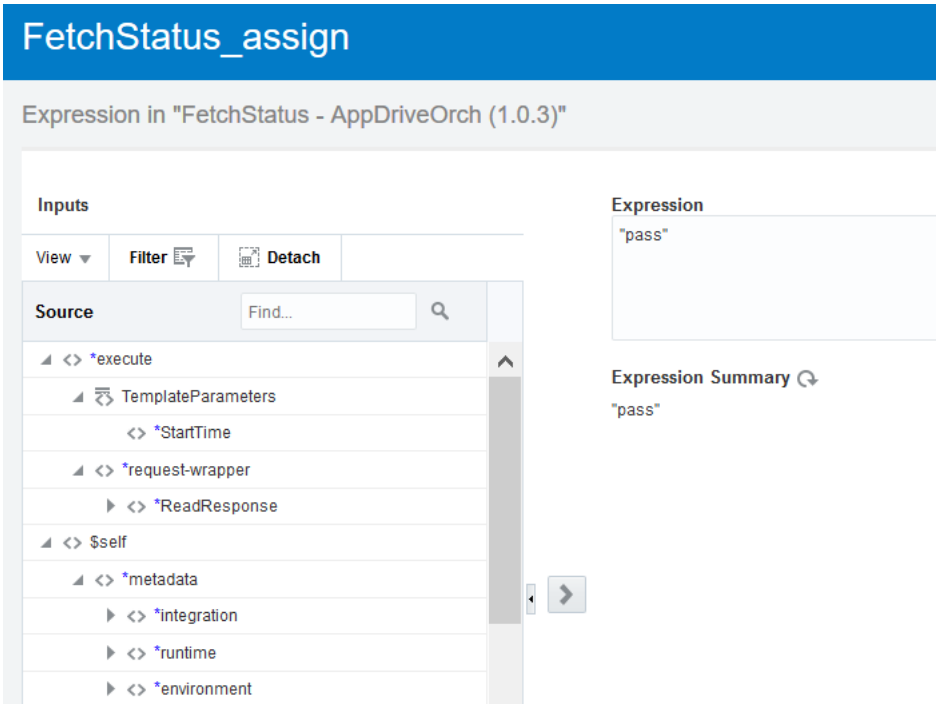
You can assign values to scalar variables in orchestrated integrations using the Expression Builder.

> **Note:**
>
> - Variables created inside a scope action or a looping action (for example, a for-each or while action) are not directly accessible outside the scope/looping action. To access the variables (local) outside a scope/looping action, create a global variable using an assign action above the scope/looping action. Assign the local variable to this global variable and then use it outside the scope/loop action.
> - Values cannot be assigned to other variable types, such as complex types.

1. Add an assign action to an integration in either of the following ways:
   - On the right side of the canvas, click  and drag the **Assign** action to the appropriate location.
   - Click  at the location where you want to add the assign action, then select **Assign**.
2. Enter an assignment name and optional description when prompted by the Assignments dialog, then click **OK**.
3. Enter a name for the assignment in the **Name** field, then click the **Expression Builder** icon at the far right. Assignment names are case sensitive.

   > **Note:**
   >
   > Once you navigate away from the Assign page (for example, go to the Expression Builder or close the Assign page), you can no longer change the name of the assign variable.

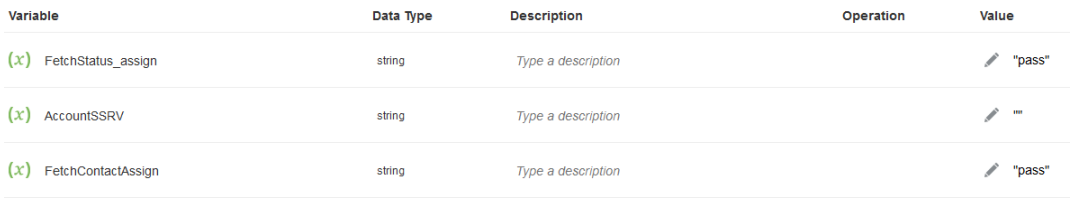
4. Build an expression, then click **Close**. See Create Routing Expression Logic in Both Expression Mode and Condition Mode. The expression is automatically saved in edit mode.



5. Click **Yes** when prompted to save your changes.

   The expression value is displayed in the **Value** field.

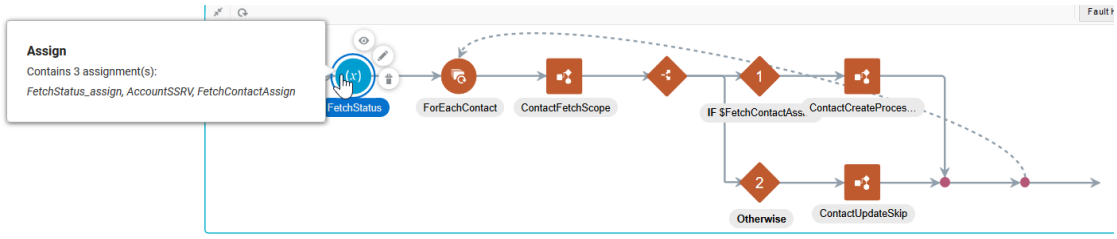
6. Click the **Add** icon to add multiple assignments to the assign action. For example:



   You can also define an assignment with a value from a previously defined assignment.

7. Click **Save**.

Variable assignments can be of greater complexity. For example, you can use assignments in switch activities and in maps. For this example, the upper branch of a switch action is taken (if **$FetchContactAssign = "failed"**), which executes the **contactCreateProcessing** scope. Otherwise, the **contactUpdateSkip** scope is executed.



You can also configure the primary tracking variable and both custom field tracking variables (update and access values) in the Expression Builder. You can map tracking variables to

output variables or create complex expressions for an assign or switch activity. The primary parameter and two custom parameters are available in the **From** palette, but only the two custom parameters are present on the **To** palette for the assign activity.

- All tracking variables are of type string (all that assignments support).
- All three tracking variable entries are present even if you choose not to model them. The name and XPath can be empty for tracking.
- Editing or deleting the tracking variables only updates the name and XPath nodes in that particular tracking variable element.
- The tracking variables have static names. Therefore, it is possible to set a tracking variable somewhere in the flow, but not initialize it with a value and a name in the Tracking dialog.
- The primary tracking variable cannot be assigned any value in between the flow.
- You cannot create a new variable with the same names as any of the statically name tracking variables.

Related Documentation
The following sections provide examples of using an assign action in an integration:

- Retrieve the Name and IDs from Account Records with the queryMore Operation in *Using the Salesforce Adapter with Oracle Integration Generation 2*
- Design a Basic Asynchronous CRUD Operation in *Using the Oracle NetSuite Adapter with Oracle Integration Generation 2*
- Use the Extract API in a Scheduled Orchestrated Integration in *Using the SAP Concur Adapter with Oracle Integration Generation 2*
- Use the Pagination Concept in an Integration in *Using the Jira Adapter with Oracle Integration Generation 2*

Loop Over Actions or Invoke Connections While a Condition is Satisfied with a While Action

The while action enables you to loop over actions or invoke connections as long as a specific condition is met. You define the condition for the while loop in the Expression Builder. The while action is available in both scheduled and nonscheduled orchestrations.

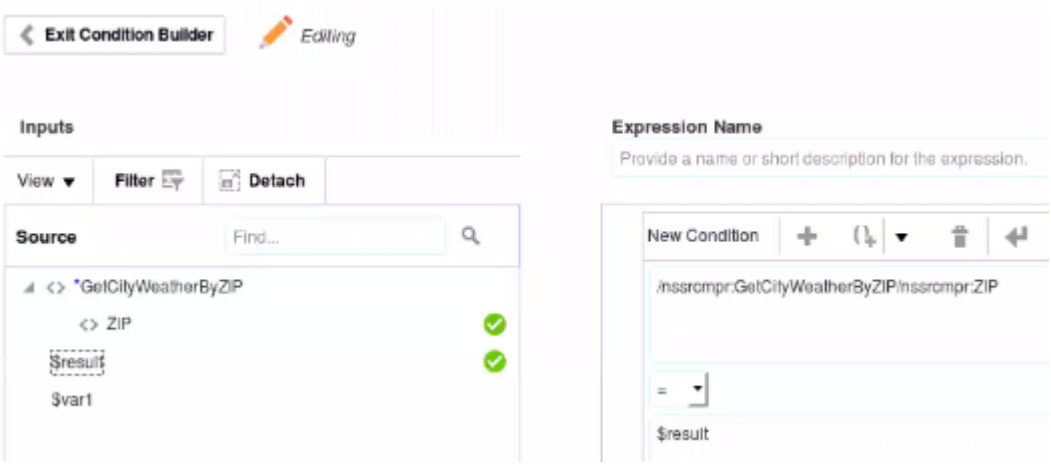
Creating a While Action

> **Note:**
>
> Variables used in while action statements can be of several types, including number, string, and boolean types.

1. Add a while action to an integration in either of the following ways:
   - On the right side of the canvas, click 
     and drag the **While** action to the appropriate location.
   - Click  at the location where you want to add the while action, then select **While**.
2. Enter a name and optional description for the while action when prompted, then click **OK**.

The Expression Builder is displayed.

3. Expand the **Source** tree to select an element. You can also add functions and operations.
4. Select an element, and click to the **New Condition** fields. For this example, while the ZIP code equals the value for **$result**, the integration loops over the condition. As soon as the condition is not met, the looping ends.

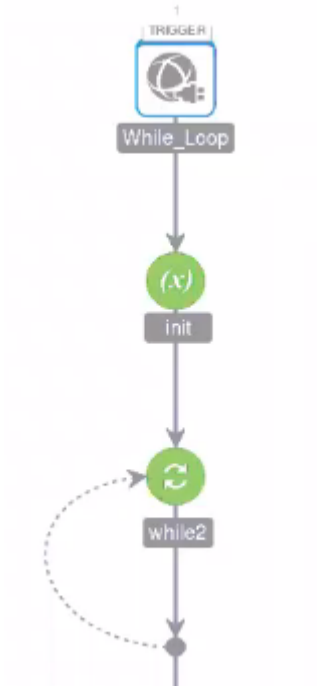


5. Click **Close**. The values of the while action are displayed in the canvas.

6. If you want to edit the name or expression, click the respective **Edit** icon. Otherwise, click

to return to the integration canvas.

The while action is displayed in the canvas. A looping arrow indicates that this action performs repetitive looping while the condition is satisfied.



7. Drag invoke connections or other actions for configuration to the **Plus** sign that is displayed inside the while action. These invoke connections and actions are executed as long as the condition of the while action is met.

> **Note:**
>
> Deleting a while action has no impact on downstream processing of the integration because the while action does not have any output. Any changes in the upstream actions in the integration that impact the while condition result in the display of a warning icon on the while action.

Tracking the Status of a While Action During Runtime

During runtime, you can track the status of the while action in the Tracking Details page through the tracking diagram and activity stream for an activated integration. This is only possible if there is a tracking instance.

1. In the left navigation pane, click **Home** > **Monitoring** > **Integrations** > **Tracking**.
2. Click the business identifier value of the integration to track.

   The integration flow (including any while actions) is displayed. Any while action failures are identified by red arrows.

3. Select **View Activity Stream** from the menu.

   Details about processing status (including any while actions) are displayed, including any failures. If failures are inside the while action, the iteration number and other details are displayed. If processing is successful, no details and counts are displayed. Any actions (including any inner while actions) inside the main while action are not recorded.

Related Documentation

The following sections provide examples of using a while action in an integration:

- Retrieve the Name and IDs from Account Records with the queryMore Operation in *Using the Salesforce Adapter with Oracle Integration Generation 2*
- Retrieve Issue Details Using the Jira Query Language (JQL) in *Using the Jira Adapter with Oracle Integration Generation 2*
- Use Pagination in an Integration in *Using the Shopify Adapter with Oracle Integration Generation 2*

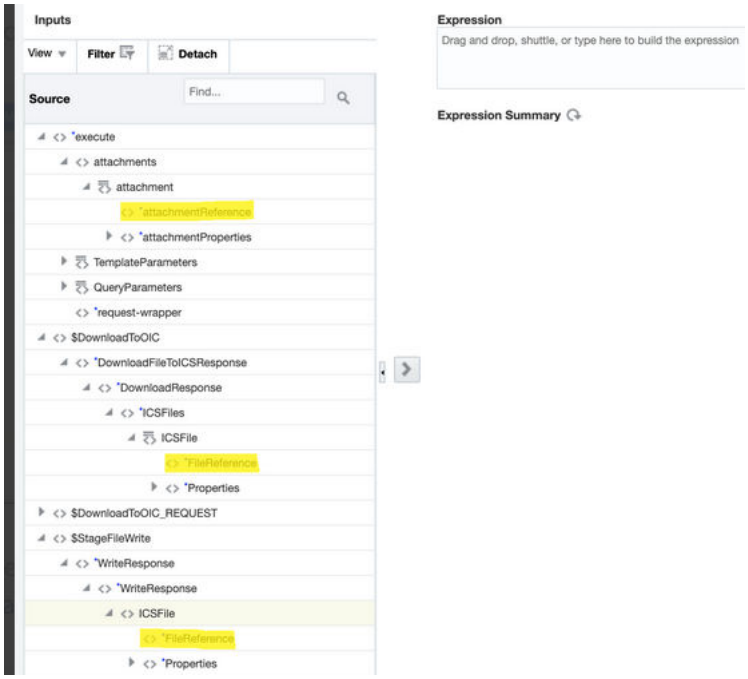
Video

Send Notification Emails During Stages of the Integration with a Notification Action

You can send a notification email to relevant users at specific points in the execution of an integration. You can set the to, from, and subject parts of an email. You can create the body part of an email using parameters defined in the Expression Builder. You can also add attachments to the email if your integration includes them. The total size limit on a notification email is 1 MB for Oracle Integration and 2 MB for Oracle Integration Generation 2. Both the email body and attachment are considered in calculating the total size.

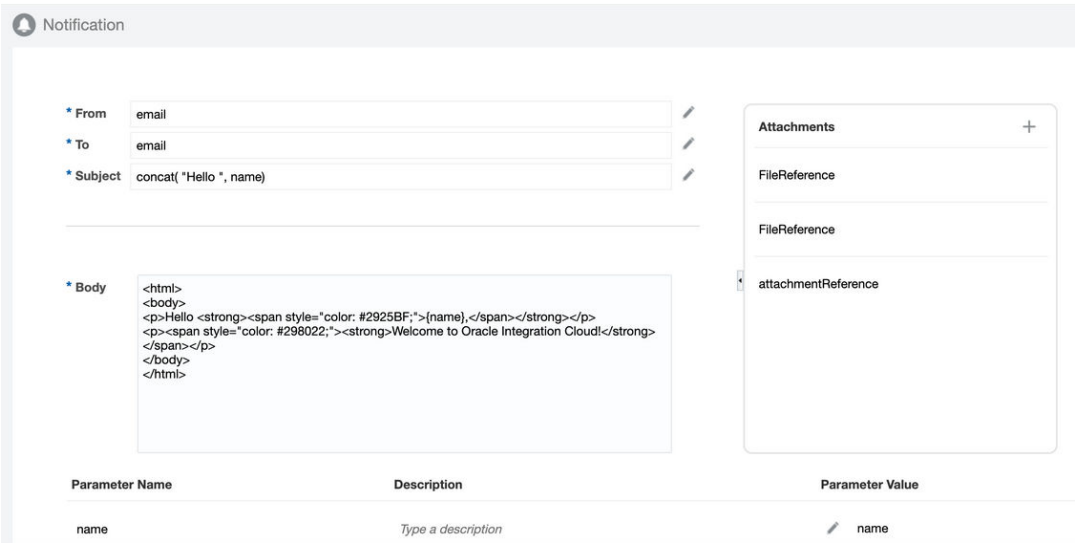
1. Add a notification action to an integration in either of the following ways:
   - On the right side of the canvas, click 
     and drag the **Notification** action to the appropriate location.
   - Click  at the location where you want to add the notification action, then select **Notification**.
2. Enter a name and optional description for the notification action when prompted, then click **OK**.
3. For the **From**, **To**, and **Subject** fields, click the **Expression Builder** icons to build the expressions to use. You can also manually enter plain text in the **Subject** field. You can provide an email address in the **From** field that is approved as the sender for service failure alerts, system status reports, and integration error reports. You configure the approval email address to use in the **From** field on the Notifications page that is accessible from **Settings** > **Notifications**. See Send Service Failure Alerts, System Status Reports, and Integration Error Reports by Notification Emails.
4. In the **Body** field, enter a message using plain text, plain HTML formatting that you create in a separate HTML editing tool and paste into this field, or parameters that you create in the table immediately below this field. After creating parameters, enter them inside `{ }` brackets.
5. To add a parameter name and value, click the **Plus** icon in the **Parameters** section.
6. Enter a parameter name and description.
7. Click the **Expression Builder** icons to define parameter values (for this example, `name` is created).
8. In the **Attachments** section, click the **Plus** icon to open a page to select an attachment file.
9. Select the attachments to add. You can edit or delete attachments once added. For this example, the integration includes three file reference attachments (highlighted in yellow) that are available for selection:

- An attachment from a REST Adapter connection
- A file reference from a stage file write operation
- A file reference from an FTP Adapter download operation



10. Click **Close**, then save your updates when prompted.

    When complete, the Notification page looks as follows for this example:



11. Click **Done**.

When the email notification is received during integration runtime, the parameter `name` is replaced with a dynamic value.

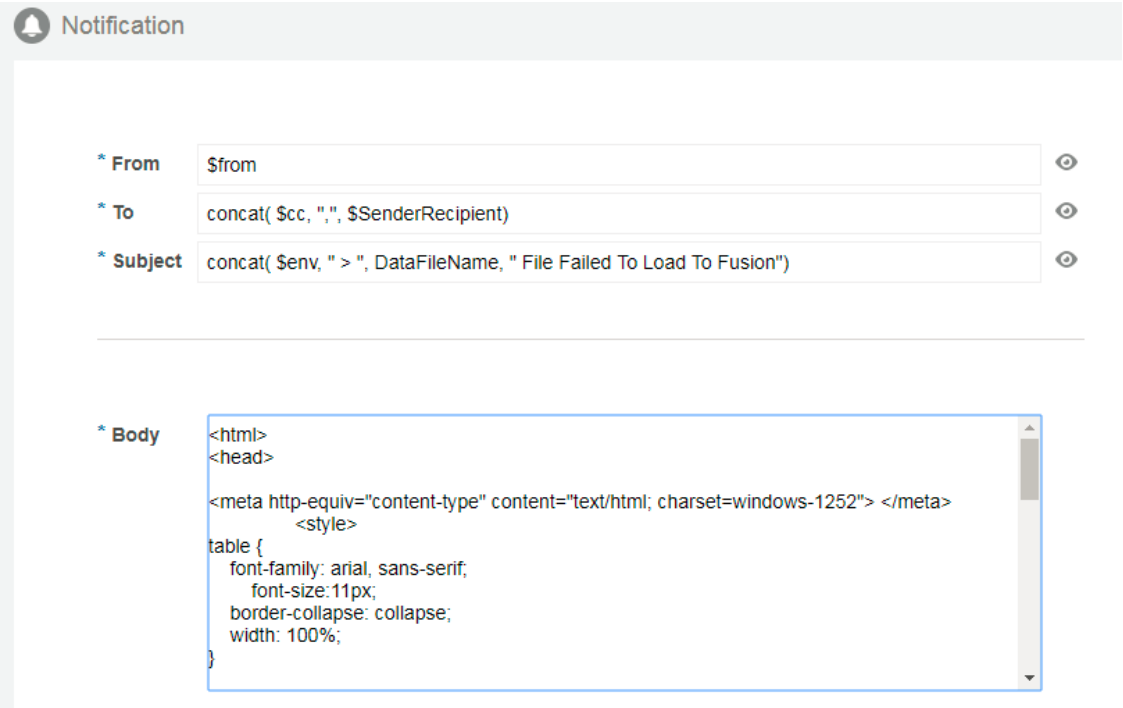
Note:

- Deleting the notification action does not impact downstream activities because a notification does not have any output. Changes in the upstream activities impact the notification when they are used either in the **From**, **To**, or **Subject** fields or in the body parameters. For example, if the `name` example used in this section is modified, the parameter assignment become invalid.
- Notification actions are treated as asynchronous actions with no failure. For example, assume you include a notification action in an integration and disable the sendmail service on your host, which prevents you from receiving an email notification. The integration instance appears as completed on the Track Instances page and there is no error message in the instance. This is the expected behavior. You can only see an issue with the instance if you open the integration instance and view the notification action.

Set the content-type in a Notification Action

You can set the `content-type` in an email body of a notification action.

Code the `content-type` as follows:



Build Complex Assignment Statements with a Stitch Action

You can incrementally build a message payload from one or more existing payloads with the stitch action. The stitch action provides an editor that enables you to assign values to variables.

The stitch action supports scalar and complex type variables, arrays, and partial and full message payloads. Complex variables are not limited to message payloads.

The stitch action differs from the mapper and assign action in the following ways:

Mapper and Assign Action Capabilities	Stitch Action Capabilities
Mapper: Only generates full message payloads. If you attempt to map into an existing message payload, a full replacement of that payload occurs.	Supports both partial and full replacement of the message payload.
Assign action: Limited to scalar type variables. Complex objects or full payloads are not supported.	Supports both scalar and complex type variables.

You can perform the following complex assignments (operations) on variables (and child elements of variables) with the stitch action:

- Append: Appends at the end of the repeating/unbounded target element, the selected element, or the value. For example, you have an existing purchase order payload containing five lines and want to add a sixth line. The stitch action enables you to append a sixth line to the existing array of lines in the purchase order.
- Assign: Places the selected value/element/attribute into the target element/attribute, overriding any existing data in the target element/attribute. For example, you want to change the current address in an existing purchase order. The stitch action enables you to change the address. You can either map fields individually or copy the address object itself.
- Remove: Removes the target element/attribute from the variable. For example, you have an existing purchase order payload and want to remove the price to enable the end point application to calculate a new price. The stitch action enables you to remove the price. For repeating/unbounded elements, all instances are removed unless a specific instance is selected by index or predicates.

Before using the stitch action, understand the use cases. See Stitch Editor Use Cases.

To build complex assignment statements with a stitch action:

1. Add a stitch action to an integration in either of the following ways:
   - On the right side of the canvas, click 
     and drag the **Data Stitch** to the location where you want to create the complex assignment statement.
   - Click  at the location where you want to add the complex assignment statement, then select **Data Stitch**.

   The right pane is refreshed to display the stitch editor.
2. Enter a name and optional description, then click **Configure**.
   The editor is refreshed to display the **Variable** list.
3. Select a variable in either of two ways:

a. Click the **Variable** list and select a variable.

b. Click

to create a global variable. Once created, the variable is available for selection from the **Variable** list.

4. From the **Variable** list, select a variable to update. Global variables and business identifier tracking variables are displayed for selection. A data element is a subcomponent of a variable. A complex variable conforms to a series of data elements.
The stitch editor is refreshed to show a list for selecting the operation to perform and the value.

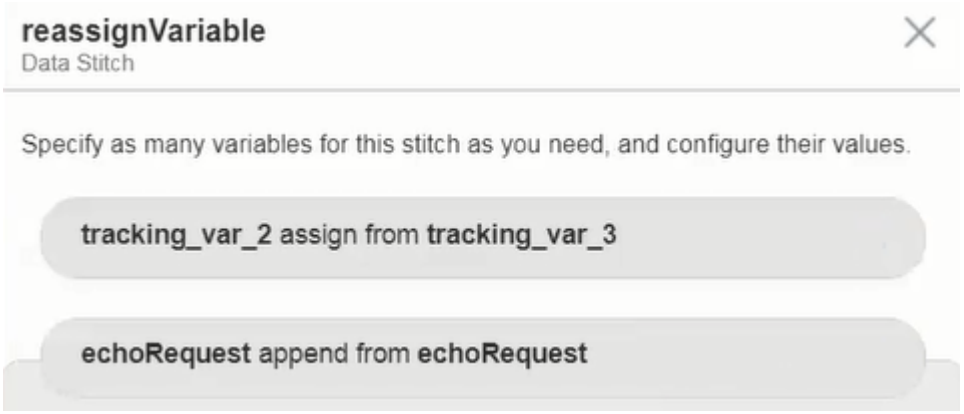
5. From the **Operation** list, select the operation to perform on the variable: **Append**, **Assign**, or **Remove**.

6. From the **Value** list, select a value for the variable. If you selected the **Remove** operation, the **Value** list is not displayed.
The page is refreshed to display a **plus** sign for creating additional complex assignment statements.

7. Click the **plus** sign if you want to create additional complex assignment statements.

8. Create as many statements as necessary.

9. When complete, click **X** in the upper right corner to save your statements and close the stitch editor.
10. Click the **View** or **Edit** icon for the stitch action in the integration to view your assignment statements.
For this example, two statements were created.



You can also define the sequence of variables to update. For example, if you want to copy an address, and then override a child element such as street, place these statements in the correct order of execution.

Delay Integration Execution for a Specified Time Period with a Wait Action

The wait action enables you to delay the execution of an integration for a specified period of time. Use this action in scheduled and asynchronous orchestrated integrations. A typical use for this action is to invoke a specific operation at a certain time. Wait actions are only available in asynchronous and fire-and-forget integrations.

Creating a Wait Action

1. Add a wait action to an integration in either of the following ways:
   - On the right side of the canvas, click  and drag the **Wait** action to the appropriate location.
   - Click  at the location where you want to add the wait action, then select **Wait**.
2. Enter the following details when prompted, then click **OK**.
   - Name of the action.
   - Optional description of the action.
   - Number of hours, minutes, and seconds to wait before executing the integration. Enter positive integer values between 0 and 59. All three fields cannot be zero. The total wait time cannot exceed six hours.
3. If you want to edit or delete the action, click it and select an option from the menu that is displayed.

Tracking the Status of a Wait Action During Runtime

During runtime, you can track the status of the wait action in the Tracking Details page through the tracking diagram and activity stream for an activated integration. This is only possible if there is a tracking instance.

1. In the left navigation pane, click **Home** > **Monitoring** > **Integrations** > **Tracking**.
2. Click the business identifier value of the integration to track.

   The integration flow (including any wait actions) is displayed.
3. Select **View Activity Stream** from the  menu.

   Details about processing status (including any wait actions) are displayed.

Related Documentation
The following sections provide examples of using a wait action in an integration:

- Use the Extract API in a Scheduled Orchestrated Integration in *Using the SAP Concur Adapter with Oracle Integration Generation 2*
- Process RaaS Reports that Support the Delivery of Data in CSV Format in *Using the Workday Adapter with Oracle Integration Generation 2*
- Export Candidate Data from Oracle Talent Acquisition Cloud (Taleo EE) in *Using the Oracle Taleo Enterprise Edition Adapter with Oracle Integration Generation 2*

Add Global Fault Handling to Orchestrated Integrations

You can add global fault handling to orchestrated integrations. This functionality enables you to direct business faults back to the caller or apply business logic before sending faults to the error handling framework. You can add fault handling to any integration type (for example, asynchronous, synchronous, or scheduled fire-and-forget (no response expected)).

Adding a Global Fault

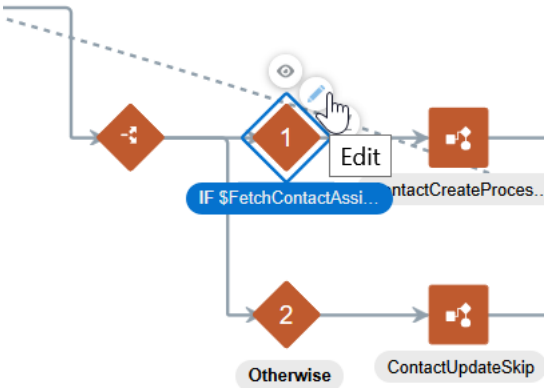
To add a global fault:

1. Design an orchestrated integration. The integration does not need to be complete. You can add fault handling at any time. However, the integration must include an invoke connection.
2. In the upper left corner of the integration canvas, select **Global Fault**.

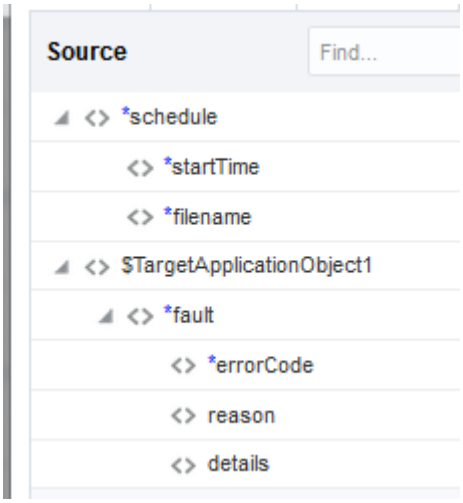
   The Global Fault Handler page is displayed. The initial trigger in your integration is automatically connected to an initial **Re-throw Fault** action that cannot be deleted. However, you can add and delete other **Re-throw Fault** actions. The **Re-throw Fault** action does not respond back to the trigger. Instead, details collected by the **Re-throw Fault** action are sent to the error handling framework.

You can add actions to design specific fault handling logic in the integration. For example, you can add **Switch**, **Stop**, or additional actions, as needed.

3. Click the upper branch, then select the **Edit** icon.



The Expression Builder is displayed. In the **Source** tree, the schedule (trigger) and target fault handling information are both displayed.



> **Note:**
>
> The Expression Builder includes the following functions under **Functions** > **Integration Cloud** for designing fault handling:
>
> - `getFaultAsString` (returns the fault as a string value)
> - `getFaultAsXML` (the fault as an XML element)
> - `getFaultedActionName` (returns the fault of the action)
> - `getFaultName` (returns the fault name)
> - `getFlowId` (returns the flow ID of the integration)
>
> These functions are only available within the Expression Builder in orchestrated integrations.

4. Build an expression to capture fault handling information. For example:

   ```
   $TargetApplicationObject1/nssrcmpr:fault/nssrcmpr:details = "ERROR"
   ```

5. Drag a **Stop** action to the **Otherwise** branch of the switch action.

   The fault handling logic is now complete. For this example, if an error occurs, error details are captured and sent to the error handling framework (as indicated by the **Re-throw Fault** action). If no error occurs, the fault handling stops and nothing is sent to the error handling framework (as indicated by the **Stop** action).

6. Click **Global Fault** to return to edit mode in the integration canvas.
7. Click **Save**.
8. Complete design of the orchestrated integration.
9. Click **Save**, then click **Close**.
10. Activate the integration.

Tracking the Status of a Global Fault During Runtime

During runtime, you can track the status of global faults in the Tracking Details page through the tracking diagram and activity stream for an activated integration. This is only possible if there is a tracking instance.

1. In the left navigation pane, click **Home** > **Monitoring** > **Integrations** > **Tracking**.
2. Click the business identifier value of the integration to track.

   Global fault handling is only invoked if there is a failure in the integration flow. Failures are identified by red arrows.

3. Select **View Activity Stream** from the  menu.

   Details about where the failure occurred and the global fault handler being triggered are displayed.

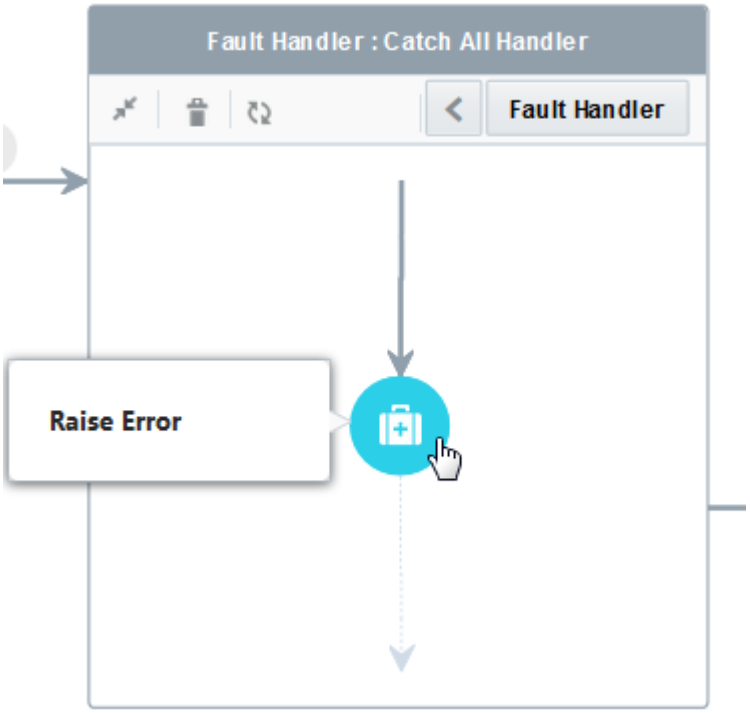
   If the global fault handler successfully handled the error, the integration is displayed as **COMPLETED** on the Track Instances page.

Catch Faults with a Re-throw Fault Action

You can send failed messages to the error hospital for further analysis with a re-throw fault action. If the integration contains a defined global fault, the error captured by the re-throw fault action is sent through the global fault and onto the error hospital for analysis. If no global fault is defined, the fault is sent directly to the error hospital for analysis. The re-throw fault action can only be placed inside the fault handler section of a scope action. The re-throw fault action operates as a catch all block and is processed if a fault is thrown by an invoke action in the scope. However, the re-throw fault action does not have a specific catch named for it. The following example describes how to define a re-throw fault action in a scope action.

To catch faults with a re-throw fault action:

1. Add a re-throw fault action to an integration in either of the following ways:
   - On the right side of the canvas, click 
     and drag the **Re-throw Fault** action to the appropriate location.
   - Click  at the location where you want to add the re-throw fault action, then select **Re-throw Fault**.
2. Create an integration that includes a scope action. See Manage a Group of Actions and Fault Handlers with a Scope Action.
3. Click **Fault Handler** on the right side of the scope.
4. On the right side of the canvas, click **Actions** to expand the panel.
5. Drag the **Re-throw Fault** icon to the **plus** sign in the scope action.



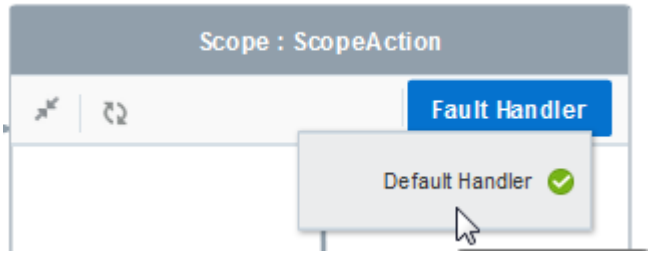
Any faults captured by this action are passed to the error hospital for analysis. Because of this fault, the integration flow is terminated.

6. Click the **<** icon to the left of **Fault Handler** to return to the scope action.

   The **Re-throw Fault** icon is hidden.

7. Design additional logic in the scope action.

8. To return to the **Re-throw Fault** icon, click **Fault Handler**, then select **Default Handler**. The green check mark indicates that this fault handler is defined in the scope action.



Throw Faults with a Throw New Fault Action

You can create and throw your own faults in an integration with a throw new fault action. During configuration of this action, you define the condition under which to throw the fault and the point in the integration at which to throw the fault. You can add this action at the end of a block (for example, a for-each action, switch action, and so on). Nothing can be dropped after this action in the block.

1. Add a throw new fault action to an integration in either of the following ways:

   - On the right side of the canvas, click
     
     and drag the **Throw New Fault** action to the appropriate location.
   - Click  at the location where you want to add the throw new fault action, then select **Throw New Fault**.

2. Enter a name and description of the action, then click **OK**.

   The Throw New Fault page is displayed.

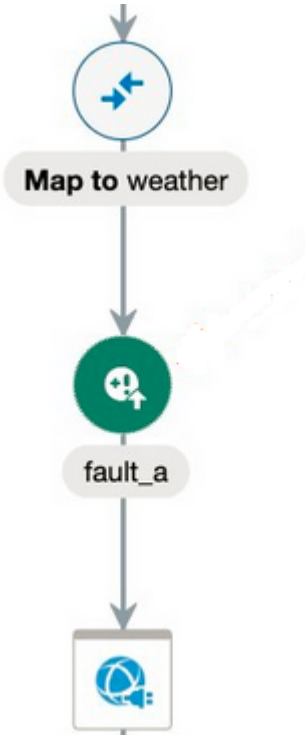
3. Define the action:

Element	Description
Code	Click the **Edit** icon to create an expression in the Expression Builder. This is a mandatory field.
Reason	Click the **Edit** icon to define a reason for the error in the Expression Builder.
Details	Click the **Edit** icon to define additional error details in the Expression Builder.
Skip Condition	Define a condition to prevent the fault from being throw in the Skip Condition version of the Expression Builder.

The throw new fault action is displayed in the integration.

If a skip condition is not defined, the throw new fault action is displayed with a dashed line connecting to the next action. This signifies that the action continues processing only after executing the fault.

If a skip condition is defined, the line connecting to the next action is solid. The solid line indicates that it is possible that the execution of the integration bypasses the fault and goes straight through to the next action.



Deleting the throw new fault action has no impact on downstream activities because this error does not have any output. Any changes to upstream activities triggers a throw new fault action validation because both the **Code**, **Reason**, and **Details** fields in the Throw New Fault page can point to flow input or upstream outputs.

Process Files in Scheduled Integrations with a Stage File Action

You can use the stage file action to process files or file references in scheduled integrations. The stage file action can process each file downloaded by the FTP Adapter. 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 stage file action can also read (and remove any trailer) and unzip referenced files in a staged location. The stage file action is similar in functionality to adapters. However, unlike adapters, you do not need to create a connection to use the stage file action. The stage file action has no credential properties and security policies. The stage file action also differs from the File Adapter and FTP Adapter in that it provides the ability to define a file format for read and write operations. For the stage file action to process or act upon files and attachments, they must be available in Oracle Integration. Make the files available in Oracle Integration either using the download operation in the FTP Adapter or consuming the SOAP/REST APIs that return multipart or MTOM attachments. Oracle Integration flows exposed as REST endpoints using the REST Adapter and exposing an interface to accept multipart attachments also automatically stage files in Oracle Integration when the requests are posted to the endpoint.

For example, you may include a stage file action in an integration as follows:

- Configure an FTP Adapter with the following settings:
  - **Download File** operation
  - **Unzip the File** option
  - Input directory and download directory path

- Because the ZIP file may contain multiple files, configure a for-each action below the FTP Adapter to iterate over repeated elements.
- To read each file from the input directory, configure a stage file action below the for-each action to read each file from the input directory as follows:
  - In the Expression Builder, specify the file name and directory from which to read the file.
  - Specify the schema file by loading a comma-separated value (CSV) file that contains the data structure.

> **Note:**
>
> - Do not use special characters in schema names (such as `#`). Otherwise, integration activation fails.
> - The stage file action only supports the `.zip` file format. For example, if the input file is `.gz` format, Oracle Integration cannot unzip or read the contents of the file.

Process File References

You can specify a file reference to process when you select to read the entire file, read a file in segments, or unzip a file.

This feature provides the following benefits:

- Processes upstream operations that provide a file reference. For example, a REST Adapter connection enables you to download an attachment into an attachment folder. The REST Adapter provides a file reference, but does not provide a file name or directory. The stage file action provides both these options.

  The following operations provide file references:

  - Attachment reference (REST Adapter attachments)
  - Stream reference (REST Adapter invoke response)
  - MTOM (for a SOAP Adapter invoke connection response)
  - FileReference (for an FTP Adapter)
- Provides a prerequisite to process encrypted content. This means that other adapters do not need to duplicate the decryption operation.
- Encrypts content as a post-processing operation. Therefore, other adapters do not need to duplicate the encryption operation.

Creating Local Files

Note the following details about local file behavior:

- Local Oracle Integrationfolders can only be created using a stage file write operation from within the integration.
- Within the integration scope, the file is available for further processing.
- Using stage file operations such as read, write, and others enables you to read the contents in the scope in which the file is available.
- The file is not visible outside the scope in which it was created.

- You can use stage file write-related variables in mapping operations to point to this virtual file.
- Once the integration moves outside the scope of file visibility, the local file is deleted.

Configure a Stage File Action

You can configure a stage file action with the Configure Stage File Action wizard.

Restrictions on Using Stage File Action Operations with the File/Attachment Features of the Connectivity Agent

When configuring the stage file action in the Configure Stage File Action wizard, note that there are restrictions on using some operations with the connectivity agent.

- List File operation: Files uploaded through the connectivity agent are not available with the List File operation.
- Read Entire File and Read File in Segments operations: Files uploaded through the connectivity agent can only be read with a file reference.
- Zip File operation: Files uploaded through the connectivity agent are not available with the Zip File operation.
- Unzip File operation: Files uploaded through the connectivity agent can only be unzipped using a file reference.

Start the Configure Stage File Action Wizard

> **Note:**
>
> Stage file operations cannot share a schema root element that defines the record structure if they are working with different file formats and record structures. Either the target namespace or the root element name must be unique while parsing different file formats and records structures.

1. Add a stage file action to an integration in either of the following ways:
   - On the right side of the canvas, click  and drag the **Stage File** action to the appropriate location.
   - Click  at the location where you want to add the stage file action, then select **Stage File**.
2. Complete the fields on the Basic Info page, and click **Next**.

   The Configure Operation page is displayed.

- Configuring Stage File Operations
- Defining the Schema File
- Defining the Schema Format
- Reviewing the Stage File Action in the Integration Canvas

Configuring Stage File Operations

The Configure Operation page enables you to define the file operations to perform. You can select to list a file, read an entire file, read files in chunked sizes, unzip a file, write a file, or zip a file, You use the Expression Builder to build the specific details of your operation. The Expression Builder shows all upstream sources (including assignments, for-each actions, invoke connections, and so on) for you to specify these details. You can select to encrypt and decrypt files with a Pretty Good Privacy (PGP) authentication key that you configured on the Upload Certificate page. See Upload an SSL Certificate.

1. From the dropdown list, select the stage file operation to perform. The page refreshes to display fields appropriate to your selection. There are restrictions when using the connectivity agent with some stage file action operations. See Restrictions on Using Stage File Action Operations with the File/Attachment Features of the Connectivity Agent.
   - Read Entire File
   - Read File in Segments (for chunking files)
   - Write File
   - Unzip File
   - Zip File
   - Encrypt File
   - Decrypt File
   - List File

Read Entire File

Table 4-1 - Read Entire File

Property	Description
Configure File Reference	• **Yes**: Select to process an upstream operation that provides a file reference. Once selected, you specify the file reference. • **No**: Select to process a file name.
Specify the File Reference (Appears if you select **Yes** for the **Configure File Reference**.)	Click the **Expression Builder** icon to specify a file reference.
Specify the File Name (Appears if you select **No** for the **Configure File Reference**.)	Click the **Expression Builder** icon to build an expression to specify the file name (for example, `/compress:schedule/compress:start Time`). **Note**: The file size must be less than 10 MB. For files greater than 10 MB, use the **Read File in Segments** operation.
Specify the Directory to read from	Click the **Expression Builder** icon to build an expression to specify the directory from which to read files.
Remove Trailer	Select to not remove the trailer, to remove the last row, or to remove the last *n* rows.
Decrypt	Click the check box, then select the private key of the target location to use to decrypt the file.

Read File in Segments

This option enables you to read files in segments (chunks). Chunking files enables large files to be processed, one logical chunk at a time. A logical chunk (segment) is retrieved from a huge file, enabling the file to stay within memory constraints.

You can also read large XML files containing repeating elements and multiple namespaces. A use case is provided. See Read Large XML Files Containing Multiple Namespaces.

Note:

- If you select the **Read File in Segments** operation, you cannot specify an opaque or JSON schema. If you do, you receive the following runtime error:

```
NXSD has infinite loop.
Flow has bad NXSD or bad input file which is causing infinite
loop.
Either NXSD is not designed properly or input file is not
compatible with
NXSD. Processing of file at stage read was not advancing and
looping at same
location in input file.
Please download ICS flow and review NXSD file or inspect input
file to ensure
there are no infinite loop.
Stage Read Failed
```

Table 4-2 - Read File in Segments

Property	Description
Configure File Reference	• **Yes**: Select to process an upstream operation that provides a file reference. • **No**: Select to process a file name.
Specify the File Reference (Appears if you select **Yes** for the **Configure File Reference**.)	Click the **Expression Builder** icon to specify a file reference.
Specify the File Name (Appears if you select **No** for the **Configure File Reference**.)	Click the **Expression Builder** icon to build an expression to specify the file name. The **Read File in Segments** option creates a stage file action that includes a scope part. This enables you to drag actions inside the scope (such as for-each actions, additional stage file actions, and others) for more complex scenarios.
Specify the Directory to read from	Click the **Expression Builder** icon to build an expression to specify the directory from which to read.

Table 4-2 (Cont.) - Read File in Segments

Property	Description
Segment Size	Chunking files enables large files to be processed, one logical chunk at a time. A logical chunk (segment) is retrieved from a huge file, enabling the file to stay within memory constraints. **Note**: • This field is not displayed when you add a new stage file action to an integration. The segment size defaults to 200 records and cannot be changed. • This field is only displayed in existing stage file actions in which a segment size of other than 200 records is specified. For these scenarios, you can specify a value between 10 and 200 records.
Process Sequentially	Select to read the segments sequentially.
Remove Trailer	Select to not remove the trailer, to remove the last row, or to remove the last *n* rows.

Write File

> **Note:**
>
> When writing to a stage file, there is no built-in support for headers. You must create headers by explicitly writing them to the file.

Table 4-3 - Write File

Property	Description
Specify the File Name	Click the **Expression Builder** icon to build an expression to specify the file name.
Specify the Output Directory	Click the **Expression Builder** icon to build an expression to specify the output directory.
Append to Existing File	Optionally select to append records to the existing file. **Note**: If you append JSON or XML content, the final file is invalid XML or JSON.
Encrypt	Click the check box, then select the public key of the target location to use to encrypt the file.

Zip Files

Table 4-4 - Zip Files

Property	Description
Specify the File Name	Click the **Expression Builder** icon to build an expression to specify the file name.

Table 4-4 (Cont.) - Zip Files

Property	Description
Specify the Directory to zip	Click the **Expression Builder** icon to build an expression to specify the directory to ZIP.
Specify the Output Directory	Click the **Expression Builder** icon to build an expression to specify the output directory in which to write a ZIP file.

Unzip File

Table 4-5 - Unzip File

Property	Description
Configure File Reference	• **Yes**: Select to process an upstream operation that provides a ZIP file reference. Once selected, you specify the file reference and the directory in which to unzip the file. • **No**: Select to process a ZIP file.
Specify the File Reference (Appears if you select **Yes** for the **Configure File Reference**.)	Click the **Expression Builder** icon to specify a ZIP file reference.
Specify the Zip File Name (Appears if you select **No** for the **Configure File Reference**.)	Click the **Expression Builder** icon to build an expression to specify the ZIP file name to read.
Specify the Zip File Directory	Click the **Expression Builder** icon to build an expression to specify the directory in which to unzip the file.
Specify the Directory to Unzip	Click the **Expression Builder** icon to build an expression to specify the directory in which to unzip files.

Encrypt File

Table 4-6 - Encrypt File

Property	Description
Specify the File Reference	Click the **Expression Builder** icon to specify a file reference.
Specify the File Name	Click the **Expression Builder** icon to build an expression to specify the file name.
Specify the Output Directory	Click the **Expression Builder** icon to build an expression to specify the output directory.
Specify PGP Key to encrypt file	Select the private key of the target location to use to encrypt the file. You can encrypt a file up to 1 GB in size. When using the mapper, note that encrypt is visible as an element for mapping.

Decrypt File

Table 4-7 - Decrypt File

Property	Description
Specify the File Reference	Click the **Expression Builder** icon to specify a file reference.
Specify the File Name	Click the **Expression Builder** icon to build an expression to specify the file name.
Specify the Output Directory	Click the **Expression Builder** icon to build an expression to specify the output directory.
Specify PGP Key to decrypt file	Select the private key of the target location to use to decrypt the file. You can decrypt a file up to 1 GB in size. When using the mapper, note that decrypt is visible as an element for mapping.

List File

Table 4-8 - List File

Property	Description
Specify the Directory to List Files from	Click the **Expression Builder** icon to build an expression to specify the directory from which to list files.
Specify the File Pattern to use	Specify the pattern of the file name to transfer to the output directory. Click the **?** icon for the list of supported patterns.
List Files Recursively	Select to list the files recursively.

2. When complete, click **Next**.

   The Schema Options page is displayed if you selected a read or write stage file operation.

Defining the Schema File

The Schema Options page enables you to define a schema. This page is displayed if you selected a read or write stage file operation.

1. Provide the following details.

Property	Description
Do you want to specify the structure for the contents of the file for the payload?	Select **Yes** to define a schema format to use for the files to transfer. Select **No** if a schema is not required and you want to send opaque files (for example, a GIF or PNG file).

Property	Description
Which one the following choices would be used to describe the structure of the file contents?	Select an option: • **Sample delimited document (erg. CSV)**: Select to create a new schema from a CSV file. On a subsequent page of this wizard, you are prompted to select the CSV file from which to create the schema. • **XML schema (XSD) document**: Select to create a new schema from an XML schema (XSD) file. You can upload a ZIP file with nested XML schema files. • **Sample XML document (Single or No Name Space)**: Select to create a new schema from a sample XML file with a single or no name space. • **Sample JSON document**: Select to create a new schema from a JSON file. • **EDI document**: Select to create a new schema from an EDI X12 document, then select the character encoding that the imported EDI X12 document is expected to use.

2. When complete, click **Next**.

Defining the Schema Format

Based on your selection on the Schema Options page, the Format Definition page enables you to select the file from which to create a schema. This page is displayed if you selected a read or write stage file operation.

1. Follow the instructions below based on the schema option you selected.
   - **Sample delimited document (e.g. CSV)** (Table 7)
   - **XML schema (XSD) document** (Table 8)
   - **Sample XML document (Single or No NameSpace)** (Table 9)
   - **Sample JSON document** (Table 10)

Table 4-9 - Sample Delimited Document (e.g. CSV)

Element	Description
• **Select the Delimited Data File**	Select the delimited comma-separated value (CSV) file from which to create the schema file. The content of the file is then displayed at the bottom of the page. This field appears if you selected to create a new schema on the Basic Info page of the wizard.
• **Enter the Record Name**	Enter the record name. This becomes the parent element in the created schema file for the record names selected as column headers from the CSV file.
• **Enter the Recordset Name**	Enter the recordset name. This becomes the root element of the created schema file.

Table 4-9 (Cont.) - Sample Delimited Document (e.g. CSV)

Element	Description
• **Select the Field Delimiter**	Select one of the following supported file delimiter options: • Single space • Comma • Semicolon • Tab • Pipe (for example, Name|City|Country)
• **Character Set**	Select a character set. The selected value will be used as the encoding format while reading the sample data file.
• **Optionally Enclosed By**	This value causes occurrences of the selected delimiter to be ignored during processing. For example, when processing the following record: `Fred,"2 Old Street, Old Town,Manchester",20-08-1954,0161-499-1718` If the selected **Field Delimiter** is "," and the **Optionally Enclosed By** value is `quot;` ("), then the value `2 Old Street, Old Town,Manchester` is treated as a single record column.
• **Use the First Row as Column Headers**	Displays by default the first row of the selected CSV file as the column headers.
• **Detach**	Select to edit the CSV file in a separate window.
• **Mark All As Optional**	Select to mark elements as optional in the schema file. By default, all elements are mandatory. You can also select the data type (for example, string, byte, integer, and so on) to use for each column in the table and mark specific elements as optional. While this option enables you to select all elements as optional, you must have at least one mandatory element to validate this page. This check box provides a convenient method to select the majority of elements as optional.

Table 4-10 - XML Schema (XSD) Document

Element	Description
• **Select a New File**	Select an existing XML schema file or schema archive file from the file system.
• **Selected File Name**	Displays the selected schema file name.
• **Select the Element Name**	Select the schema element. This field is displayed after the XML schema file is selected. The element name is treated as the root element in the uploaded schema file.
• **Select Repeating Batch Element**	Click the **Expression Builder** icon to build an expression to identify the repeating node in the schema to support payload chunking.

Table 4-11 - Sample XML Document (Single or No NameSpace)

Element	Description
• **Select a New File**	Select a sample XML document from the file system. The file must contain no namespace or a single namespace.
• **Selected File Name**	Displays the selected schema file name.
• **Select the Schema Element**	Select the schema element. This field is displayed after the sample XML file is selected. The element name is treated as the root element in the uploaded schema file.
• **Select Repeating Batch Element**	Click the **Expression Builder** icon to build an expression to identify the repeating node in the schema to support payload chunking.

Table 4-12 - Sample JSON document

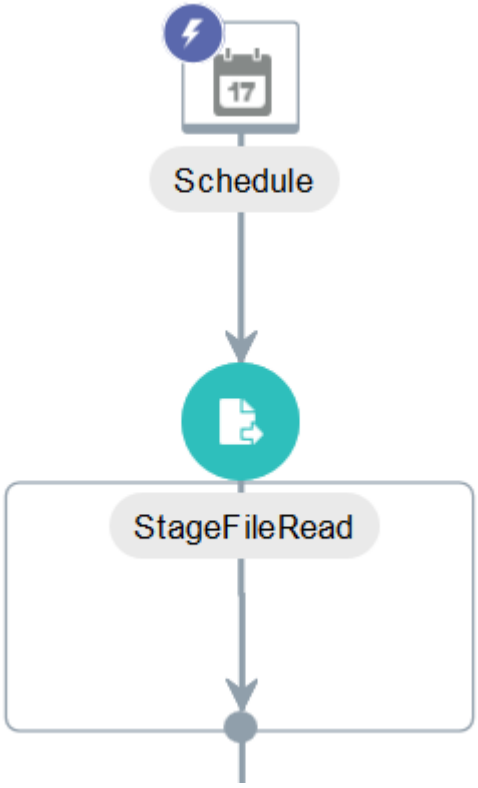
Element	Description
• **Select a New File**	Select a sample JSON document from the file system.
• **Selected File Name**	Displays the selected schema file name.
• **Select the Schema Element**	Select the schema element. This field is displayed after the JSON file is selected. The element name is treated as the root element in the uploaded schema file.

2. Complete the fields, and click **Next**.
3. Review your selections on the Summary page, and click **Done**.

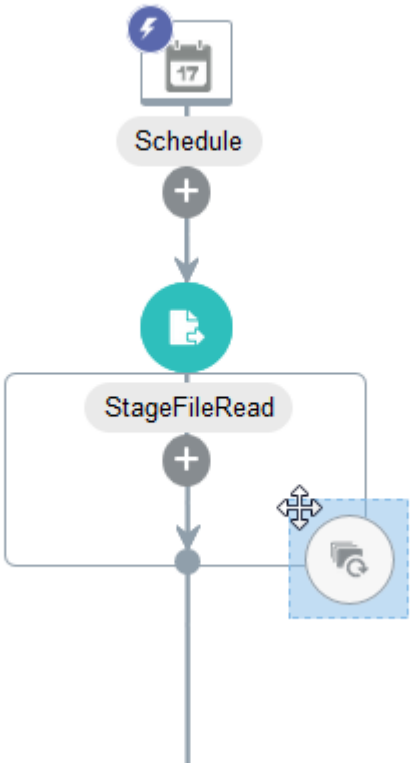
Reviewing the Stage File Action in the Integration Canvas

Once design is complete, the stage file action is displayed in the integration canvas.

- If you designed a stage file action with a **Read File in Segments** operation, a scope portion is created. Place your cursor over the icon to display a message indicating that segmentation (chunking) is enabled.

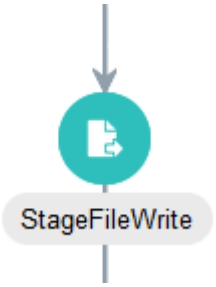


You can drag additional actions into the scope part of the stage file action to perform tasks. For example, you can add a for-each action to the stage file action to process the segmented chunks of large files one record at a time. You can also add child stage file actions to perform further processing on each of the chunks. However, you cannot configure additional chunking on the child stage file action.



> **Note:**
>
> If a variable is declared outside of a loop of a stage file action that was configured with the **Read File in Segments** operation and updated within the stage file action loop, the last updated value for that variable is available outside of the loop. In such a case, the DOP (degree of parallelism) is set to `1` to avoid concurrency issues.

- If you designed a stage file action with an operation other than a **Read File in Segments** operation, a scope portion is not included.



Complete your integration design and configure any business identifiers for tracking fields in messages (including file storage-related parameters).

Video

Related Documentation
The following sections provide examples of using a stage file action in an integration:

- Create an Integration to Import and Process Bulk Files in *Using the FTP Adapter with Oracle Integration Generation 2*
- Process RaaS Reports that Support the Delivery of Data in CSV Format in *Using the Workday Adapter with Oracle Integration Generation 2*
- Produce Messages to an Apache Kafka Topic in *Using the Apache Kafka Adapter with Oracle Integration Generation 2*
- Use Bulk Response Operations in an Integration in *Using the Salesforce Adapter with Oracle Integration Generation 2*

Video

Log Messages with a Logger Action

You can log messages to the activity stream and diagnostic logs by adding the logger action at any point in the integration. You create a log message in the logger action that is a static message or variable-populated message in the Expression Builder.

Creating a Logger Action

To create a logger action:

1. Add a logger action to an integration in either of the following ways:
   - On the right side of the canvas, click
     
     and drag the **Log** action to the appropriate location.
   - Click  at the location where you want to add the log action, then select **Log**.
2. Enter a name and optional description, then click **OK**.

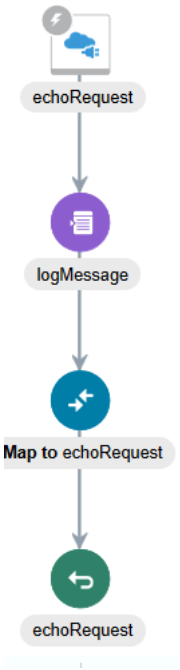
   The Logger page is displayed.
3. Select whether to always log this message or log it only when tracing is enabled. You can enable tracing when you activate an integration. See Activate an Integration.
4. Specify a static or variable-populated message to be displayed in the activity stream and diagnostic logs. Click the **Expression Builder** icon to create a message that includes variables. For this example, the logger is being configured to indicate that a specific file name is being read from an FTP server location.

   ```
   concat("Filename is: ",fileName)
   ```

5. Drag and configure additional **Logger** icons into the integration, as needed. For this example, the another logger is being configured to indicate that a file has been uploaded to a directory location.

   ```
   concat("The file ",fileName, "has been uploaded to ",directory)
   ```

6. When complete, click **Close**. For this example, a log action is included in the integration.



Tracking the Status of a Logger Action During Runtime

During runtime, the messages in the loggers are written to the activity stream and diagnostic logs.

1. In the left navigation pane, click **Home** > **Monitoring** > **Integrations** > **Tracking**.
2. Select **View Activity Stream** from the menu.

   Details about processing status are displayed.
3. In the left navigation pane, click **Dashboards**.
4. Select **Diagnostic Logs** from the **Logs** list.
5. Open the zip file and view the log messages you created.

Add a JavaScript Action

You can add JavaScript functions to the integration.

Creating a JavaScript Action

To create a JavaScript action:

> **Note:**
>
> Note the following restrictions when using the JavaScript action:
>
> - The JavaScript action has a time out threshold of 15 seconds. Any JavaScript function that executes for more than 15 seconds fails with a time out error.
> - JavaScript functions are not allowed to make outbound calls to external services. Any outbound calls are blocked and ultimately fail.
> - Network, disk access, or thread access functions are not supported.

1. Add a Javascript action to an integration in either of the following ways:
   - On the right side of the canvas, click  and drag the **Javascript** action to the appropriate location.
   - Click  at the location where you want to add the Javascript action, then select **Javascript**.
2. Enter a name and optional description for the JavaScript action when prompted, then click **OK**.
3. Click the **+Function** button.

   The Select a Function dialog appears.
4. Click a function, then click the **Select** in the function's row.

   The configuration page is displayed. It shows the details of the selected function including the input and output parameters.
5. Click the  icon in the **Value** column to use the Expression Builder to configure the input parameters.
6. Click **Validate** in the title bar to validate the parameters.
7. Click **Close** in the title bar to close the page.

Tracking the Status of a Javascript Action During Runtime

During runtime, you can track the status of the JavaScript action in the Tracking Details page through the tracking diagram and activity stream for an activated integration. This is only possible if there is a tracking instance.

1. In the left navigation pane, click **Home** > **Monitoring** > **Integrations** > **Tracking**.
2. Click the business identifier value of the integration to track.
3. The integration (including any JavaScript actions) is displayed. Any JavaScript action failures are identified by red arrows.
4. Select **View Activity Stream** from the  menu.

   Details about processing status (including any JavaScript actions) are displayed, including any failures.

See Use Libraries to Manage Integration Functions.

Add Placeholder Notes with a Note Action

You can add placeholder notes similar to sticky notes to an integration. For example, you have not yet defined an invoke connection and want to add a placeholder note in the integration indicating that you plan to define the invoke connection later. Another integration developer reads that note and may add the invoke connection or the note reminds you to add the invoke connection at a later time when you again work on the integration.

The note action is a design-time feature that does not impact runtime. Any changes to upstream or downstream actions in your integration do not impact the note action.

To add a note action:

1. Add a note action to an integration in either of the following ways:
   - On the right side of the canvas, click  and drag the **Note** action to the appropriate location.
   - Click  at the location where you want to add the note action, then select **Note**.
2. Enter a name and description of the action, then click **OK**.

   The action is added to the integration.
3. Click the **Edit** icon to add your notes. You can add up to 256 characters.
4. Enter your notes, then click **OK**.
5. Hover your cursor over the icon to display the note text.

Translate an EDI Document with the B2B Action

You can translate a message to or from the Electronic Data Interchange (EDI) format in an orchestrated integration with the B2B action.

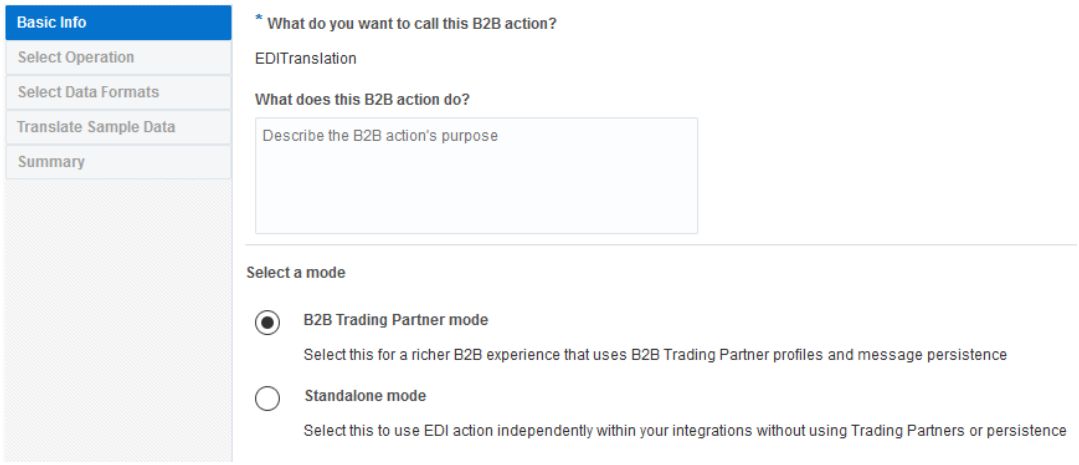
The B2B action translates an incoming EDI document into an Oracle Integration XML message and an outgoing Oracle Integration XML message into an EDI X12 document.

1. Add a B2B action to the integration in either of the following ways:
   - On the right side of the canvas, click  and drag the **B2B** icon to the appropriate location.
   - Click  at the location where you want to add the action, then select **B2B**.

   The Configure B2B Action Wizard is displayed.
2. Enter a name and optional description.
3. Select how you want to use the B2B action, then click **Next**.
   - **B2B Trading Partner Mode**: Select to include B2B trading partner profiles and message persistence in your integration.

> **Note:**
>
> This option is only available with Oracle Integration Generation 2.

- **Standalone Mode**: Select to use the B2B action independently in your integration *without* a B2B trading partner profile or message persistence.



4. If you select **B2B Trading Partner Mode**, follow these wizard steps:
   a. Select the B2B message direction for this B2B action:
      - **Inbound**: The B2B message is sent from the trading partner to the host trading partner (partner where Oracle Integration is installed).
      - **Outbound**: The B2B message is sent from the host trading partner (partner where Oracle Integration is installed) to the trading partner.
   b. Select the operation for the B2B action to perform based on the direction.

If You Selected...	Then Select an Operation...
Inbound	• **Fetch Message**: Fetches a B2B business message from the persistence store, given a message identifier. • **Translate**: Translates an inbound EDI message to an Oracle Integration message. • **Mark As Error**: Marks a B2B business message with a failed processing status.
Outbound	• **Translate**: Translates an Oracle Integration message to an outbound EDI message. • **Mark As Error**: Marks a B2B business message with a failed processing status.

   c. If you select either of the following, the Select Data Format page is displayed.
      - **Inbound** message direction and **Fetch Message** operation
      - **Outbound** message direction and **Translate** operation
      i. Select the document definition for the B2B action to handle, and click **Search** to refresh the page.

ii. Select the document standard, version, and type.

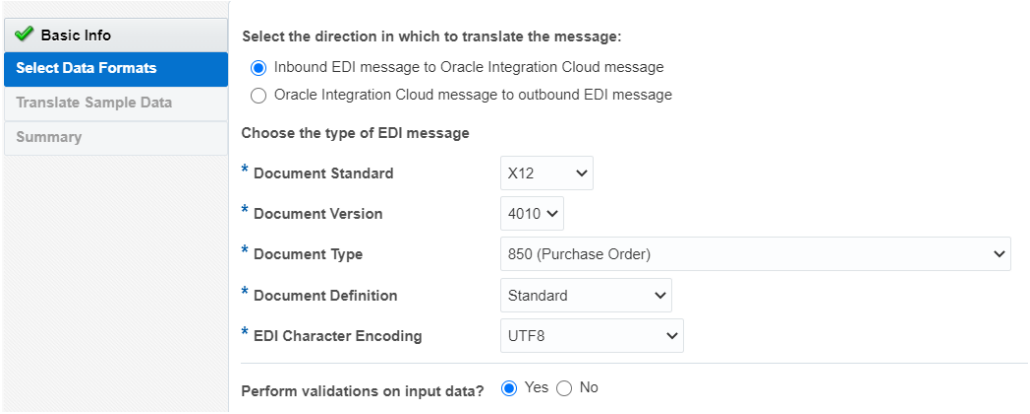
iii. Click **Next**.

d. Review your selections on the Summary page, then click **Done**.

5. If you select **Standalone Mode**, follow these wizard steps:

a. Specify the message translation and document format details.

Element	Description
Select the direction in which to translate the message	• **Inbound EDI message to Oracle Integration message**: When an integration receives an EDI document from a business partner, it is considered an inbound document (an EDI document is translated to XML). • **Oracle Integration message to outbound EDI message**: When an integration sends an EDI document to a business partner, it is considered an outbound document (an EDI document is generated from XML).
Document Standard	The document standard identifies the business protocol to follow when exchanging business documents between partners. • EDIFACT • X12
Document Version	Lists the supported versions of the selected document standard. Select the version to use.
Document Type	Select the document type (for example, purchase order, invoice, shipping notice, or others). The document types available for selection are based on the document version you selected.
Document Definition	Select the document definition (either **Standard** or a custom document definition that you created on the B2B Documents page). See Create Custom Document Definitions in *Using B2B for Oracle Integration Generation 2*.
EDI Character Encoding	Select the character encoding that the inbound EDI document is expected to use.
Perform validations on input data	• **Yes**: Validates the structure and data of an inbound EDI message. Enabling message validation has an impact on performance. If errors are found, translation does not succeed. • **No**: Errors are ignored during translation and the message is passed through in its current format.



b. Click **Next**.

c. Specify to optionally upload sample data to test that translation is successful.

Element	Description
Select sample data to upload	Click **Browse** to upload a sample. To test inbound EDI message translation, upload an EDI document. To test outbound EDI message generation, upload an XML document.
Translate	Click to translate your sample data. Output is displayed in the **Output of translation** field. Any errors are displayed in the **Error in translation** field.

d. Click **Next**.

e. Review your selections on the Summary page, then click **Done**.

Use XPath Axis and Wildcard Expressions in the Expression Builder

You can include XPath axis and wildcard expressions in actions that support the Expression Builder (for example, assign and switch actions support the Expression Builder).

1. Create an assign action or switch action and navigate to the Expression Builder.
2. In the **Expression** field for an assign or switch action, build an expression using either option:
   - Wildcard expression:

     For this example, a wildcard is entered to select all elements below `Answer`.

     ```
     /nssrcmpr:process/nssrcmpr:Answer/*
     ```

     Or, to select all elements:

     ```
     /*
     ```

   - Axis expression:

For this example, `descendant` is entered to select all descendants (child, grandchildren, and so on) of the current node. Any descendant with this ID and namespace (`mb_v1_3:ID`) is retrieved by the expression.

```
/nssrcmpr:process/nssrcmpr:Answer/descendant::mb_v1_3:ID
```

Axis expressions adhere to the following syntax:

axisname::*nodetest*[*predicate*]

The following table provides examples of axis expressions:

Syntax	Result
`child::book`	Selects all `book` nodes that are children of the current node.
`attribute::lang`	Selects the `lang` attribute of the current node.
`child::*`	Selects all element children of the current node.
`attribute::*`	Selects all attributes of the current node.
`child::text()`	Selects all `text` node children of the current node.
`child::node()`	Selects all children of the current node.
`descendant::book`	Selects all `book` descendants of the current node
`ancestor::book`	Selects all `book` ancestors of the current node.
`ancestor-or-self::book`	Selects all `book` ancestors of the current node and the current if it is a `book` node.
`child::*/child::price`	Selects all `price` grandchildren of the current node.

3. Click the **Expression Summary** icon to validate the expression.
4. When complete, click **Close**.

Use Lookups in Variable Assignments

You can create variable assignments that use lookups in the Expression Builder. You must have already created the lookup.

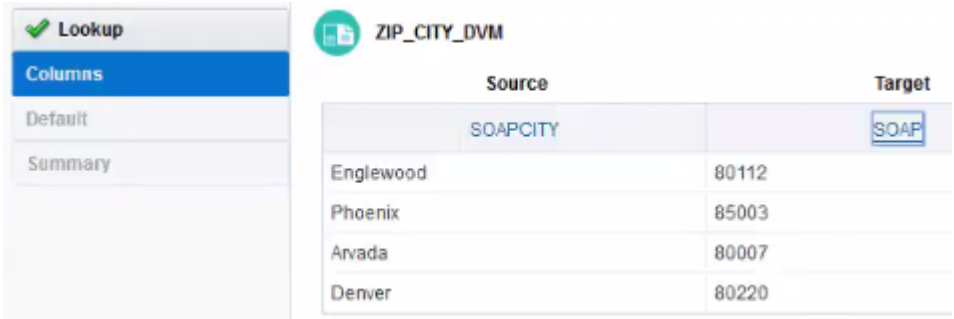
1. Create a lookup in Oracle Integration. See Manage Lookups.

   For this example, a lookup named **ZIP_CITY_DVM** is created to map the ZIP codes (using a SOAP Adapter) and the city names (using a domain name).

SOAP (Adapter)	SOAPCITY (Domain Name)
80112	Englewood
85003	Phoenix
80007	Arvada
80220	Denver

2. Create an orchestration into which a SOAP Adapter is added as the trigger.
3. Drag the **Assign** icon to the **plus** sign for the SOAP Adapter in the integration canvas.
4. Enter an assignment name and optional description when prompted by the Assignments dialog, then click **OK**.
5. Enter a name for the assignment in the **Name** field, then click the **Expression Builder** icon at the far right.
6. In the Expression Builder, expand **Functions**, then **Oracle Integration**.
7. Drag the **lookupValue** function to the **Expression** field.

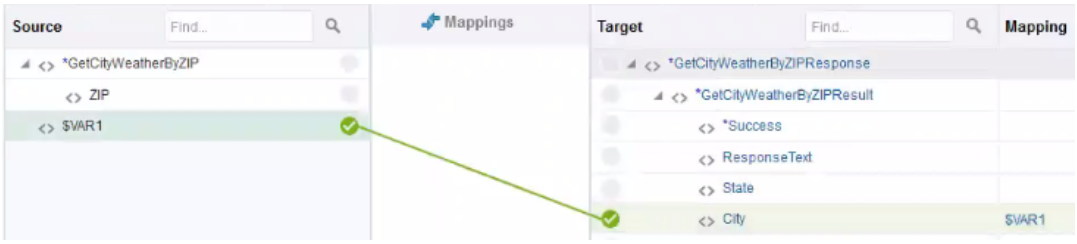
   This starts the Build Lookup Function wizard.
8. Select the lookup table, and click **Next**. For this example, the lookup created in Step 1 (**ZIP_CITY_DVM**) is selected.
9. Select the source and target columns. For example, select to pass the city to the ZIP code.



10. Complete the remaining pages of the wizard, then click **Done**.
11. In the **Source** tree, select the element to map. For this example, **ZIP** is dragged on top of **srcValue** in the **Expression** field.
12. Click **Close**.

    The completed variable assignment is displayed.
13. Click **Exit Assignments**.

14. In the orchestrated integration, click the **mapper** icon, then click the **Edit** icon.
15. Assign the variable you created to the city target element, then click **Save**.



16. Click **Close**.

Define Fault Aggregation in Parallel Processing Scenarios

Oracle Integration includes a predefined fault object for fault aggregation. This object enables you to catch and aggregate faults in the context of parallel processing in scheduled orchestrated integrations and to send these faults to an external service to define more granular error handling. To define a fault object and aggregate the faults, you must use actions that support looping (for example, for-each loops, while loops, and so on).

> **Note:**
>
> The fault object is only supported with scheduled orchestrated integrations.

Two fault aggregation operations are available for selection in scheduled orchestrated integrations:

- **append**: Adds a new fault to the end of a fault list.
- **clear**: Removes all entries in a fault list.

These options are available for selection under the following scenarios:

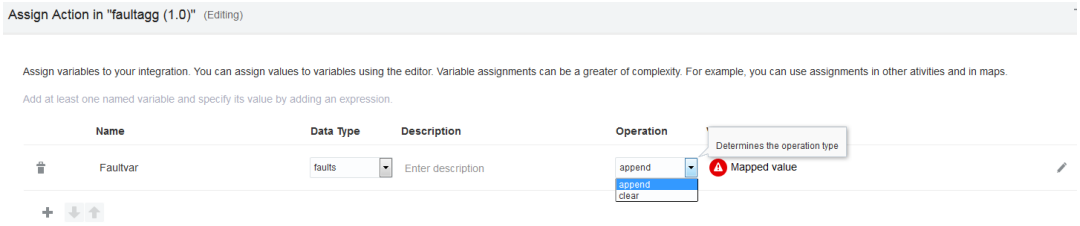
Location	Append Option Allowed?	Clear Option Allowed?
Top level (that is, outside of any looping actions)	Yes	Yes
Inside a for-each action with the **For Process items in parallel** option selected	Yes	No
Inside a for-each action loop and a while action loop	Yes	Yes
Inside a stage file action configured with the **Read File in Segments** operation	Yes	No
Inside a scope action	Yes	Yes
Inside a global fault or a named fault (that is, inside a scope fault handler)	Yes	Yes

To configure fault aggregation:

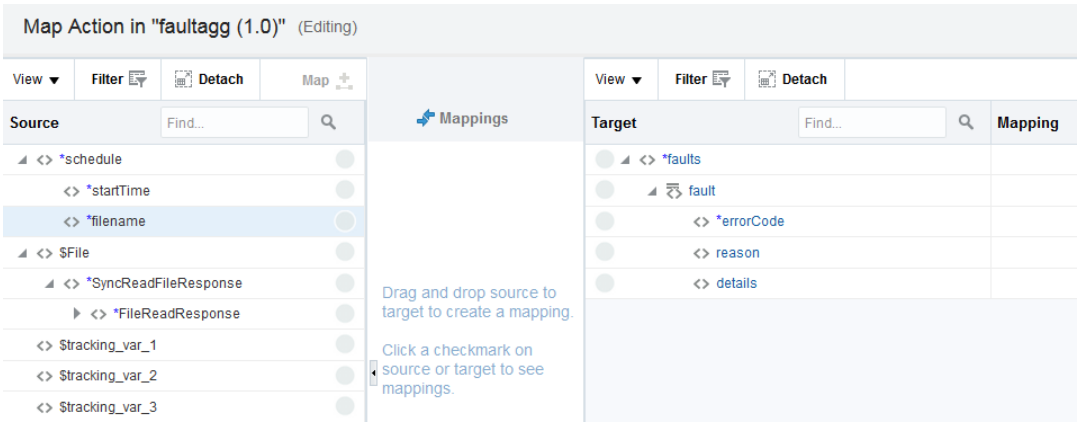
1. Created a scheduled orchestrated integration.

2. Create an aggregated fault per the scenarios supported in the above table. For example, create a **For-Each** action, then drag an **Assign** action within the loop.
3. Create an assignment and select **faults** from the **Data Type** list.

   The page is refreshed to display the **Operations** list with two options.



4. Click the **Expression Builder** icon to build the fault object mapping.



5. Complete your mapping.
6. Define an invoke connection to send the aggregated faults to an external service.

> **Note:**
>
> Changing the name, data type, or both is only allowed during initial creation of the variable while not yet assigned a value.

Assign Business Identifiers for Tracking Fields in Messages

This section describes how to manage business identifiers that enable you to track fields in messages during runtime.

Topics:

- Assign Business Identifiers
- Delete Business Identifiers

Assign Business Identifiers

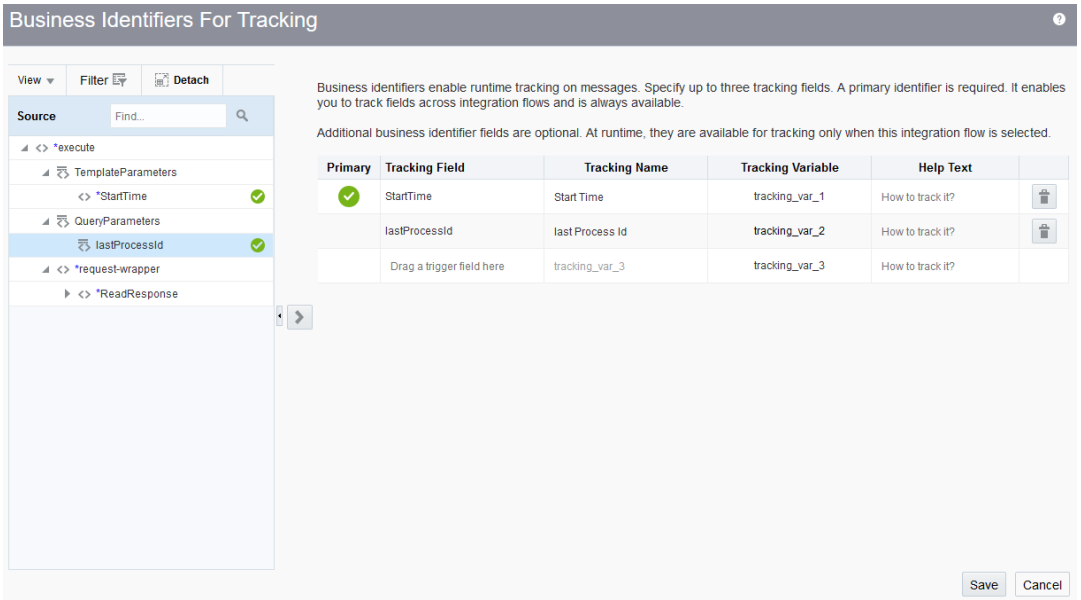
Business identifiers enable you to track payload fields in messages during runtime. You can specify up to three business identifier fields for tracking during design time. One of these fields must be selected as the primary business identifier field. The primary business identifier enables you to track fields across integration flows during runtime, and is always available. At runtime, the status of business identifiers is visible on the Track Instances page and (if integration errors have occurred) the Errors page. If you created scheduled parameters, they are available for assignment as business identifiers.

If you have not yet configured at least one business identifier tracking field in your integration, an error icon is displayed in the design canvas. See Missing a Primary Business Identifier in an Integration.

To assign business identifiers:

1. In the left navigation pane, click **Home** > **Integrations** > **Integrations**.
2. Click the specific integration to which to add business identifiers. You can only add business identifiers to integrations that are *not* active. If an integration is active, you can only view its existing business identifiers.
3. From the  menu, select **Tracking**.

   The Business Identifiers For Tracking dialog is displayed. The source payload for the selected integration is displayed on the left side. You can only assign business identifiers to fields of source payloads. You cannot assign business identifiers to fields of target payloads.
4. From the **Source** section, drag the payload field that you want to track to the **Drag a trigger field here** section. A green checkmark indicates that this is the primary business identifier. At least one identifier is required. If you only add one, it is automatically selected as the primary key. If you created scheduled parameters, they are available for assignment as a primary or secondary business identifier.



You can filter the display of source structures by clicking the **Filter** link. This enables you to filter on whether or not fields are used and on the type of field (required fields, custom fields, or all fields).

5. In the **Tracking Name** field, optionally enter a descriptive name to track during runtime (for example, `OrgId`). The name is displayed when this field is used to filter messages on the Track Instances page or (if there is an integration error) the Errors page during runtime.
6. In the **Help Text** field, optionally enter instructions to enable users to know what to enter in this field during runtime (for example, `Enter a valid organization number`). These instructions are displayed inside the empty field when it is used on the runtime Track Instances page to filter messages.
7. Click **Done**.

Delete Business Identifiers

You can delete business identifiers that track fields in messages during runtime.

To delete business identifiers:

1. In the left navigation pane, click **Home** > **Integrations** > **Integrations**.
2. Click the specific integration to which to add a business identifier. You can only add business identifiers to integrations that are *not* active. If an integration is active, you can view, but not edit, the contents of the Business Identifiers for Tracking dialog.
3. From the  menu, select **Tracking**.

   The Business Identifiers For Tracking dialog is displayed.
4. At the far right, click the **Delete** icon for the business identifier to delete. If you delete the primary business identifier, select a new one. Without a primary identifier, you cannot track fields across integration flows during runtime on the Track Instances page.

Display Errors and Warnings in an Integration

If there are errors or warnings in an integration (for example, an empty or invalid map, a missing tracking attribute, or an invalid assign or switch action), an **ERRORS** section is displayed on the left side. These errors and warnings prevent you from activating an integration. You must first resolve these issues to activate an integration.

> **Note:**
>
> - For integrations created prior to version 16.4.5, you *must* first save the integration once to enable error and warning validation functionality.
> - Error and warning validation of for-each actions is not supported.

1. Design an orchestrated integration.

   If there are errors or warnings, an **Errors** section is displayed on the right side.
2. Click the **Errors** section to display error and warning details. For this example, an invoke adapter was deleted for which mapping had previously been configured, causing the mapper to be invalidated. In addition, tracking is not configured.

3. If you return to the Integrations page, note that the status of the integration is **DRAFT**. You cannot activate integrations in the **DRAFT** state.
4. Return to the integration canvas and resolve any errors and warnings. Once these issues are resolved, the **ERRORS** section disappears.
5. Save the integration and return to the Integrations page. Note that the **DRAFT** status is replaced with **PENDING ACTIVATION**. You can now activate the integration.

5

Create Scheduled Integrations

You can schedule the running of integrations. For example, you can create an orchestrated integration that is triggered by a schedule or create a basic routing integration in which a trigger FTP Adapter reads a file and an invoke FTP Adapter writes the file. You can schedule this integration run to copy files at a date and time of your choosing. You can also define the frequency of the integration. When you create either of these types of integrations, a schedule icon is displayed with the integration entry on the Integrations page.

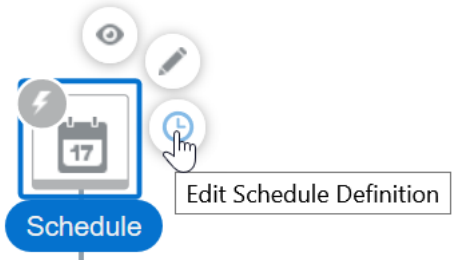
Topics:

See *Understand the FTP Adapter* in *Using the FTP Adapter with Oracle Integration Generation 2*.

Create a Scheduled Integration

This section describes how to create a scheduled integration.

1. Follow the steps in Create Integrations to create a scheduled integration.
   An empty integration canvas is displayed with a schedule icon at the top.



2. To define the schedule for running the integration, click the **Schedule** icon, then select **Edit Schedule Definition**. See Define the Integration Schedule. You can also define a schedule on an integration at a later time from the Integrations page.
3. See Create an Orchestrated Integration to design the entire flow of your scheduled integration and Assign Business Identifiers for Tracking Fields in Messages to assign business identifiers. Scheduled integration design differs from application-driven orchestrated integration design only in that you define an adapter trigger connection in an application-driven orchestrated integration. In scheduled integrations, the schedule is the trigger in the integration.

Define the Integration Schedule

You can define a schedule for running orchestrated integrations, such as the frequency at which to run the integration, whether to run the schedule based on an iCal expression, whether the schedule run should never expires, whether the schedule should have a fixed expiration date, and so on.

To define an integration schedule:

> **Note:**
>
> - You cannot separately export the schedule of an integration.
> - When the user that created and scheduled an integration is locked or deleted, the schedule does not run. Ensure that this user is not locked or deleted.

1. Go to the Integrations page.
2. Find the scheduled integration.

   These integrations are identified by a
   
   schedule trigger icon in the far left column. There are several ways in which to create a schedule on an integration.
3. If you want to first design and activate the integration and then create the schedule:
   a. Click the
      
      icon. The Activate Integration dialog box is displayed.
   b. Click **Activate and Schedule**.
4. If you want to first create a schedule for an integration that you design and activate later:

- Select **Add Schedule** from the  menu.

> **Note:**
>
> For integrations with no defined schedules, the menu shows **Add Schedule**. For integrations that have a defined schedule, the menu switches to **Schedule**.

The page for defining the schedule execution details is displayed.

> **Note:**
>
> Assume you create a schedule and generate several integration runs, then deactivate and delete the integration. If you then create and activate a new integration with the same name as the deleted integration, and go to the Track Instances page, the past runs of the deleted integration are displayed. This is by design.

5. If you want to schedule basic integration runs, click **Simple**. For this type, there is a minimum frequency limit of ten minutes. If you try to define a schedule frequency of under ten minutes, a validation error occurs.

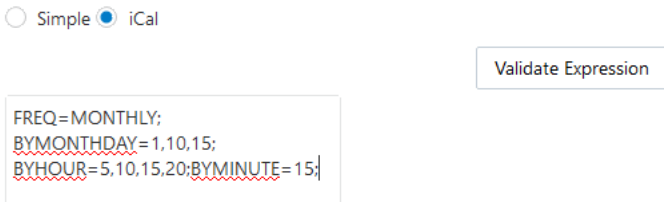
   - In the **Frequency** section for a **Simple** schedule, click the icon to display a dropdown list for selecting the frequency with which to run the integration. As you define one frequency, you can specify additional values by clicking the icon to the right of the **Frequency** section.
     - **Only Once**: This is the default selection. This selection clears all settings except for the **From** field.

       If you select this option, you cannot select **When schedule starts** as the start date. This option is disabled.
     - **Hours and Minutes**: Specify the hours and minutes at which to run the integration.
     - **Days**: Specify the days on which to run the integration.
     - **Weeks**: Specify the weeks during which to run the integration.
     - **Months**: Specify the months during which to run the integration.

6. If you want to schedule integration runs with an iCal expression, click **iCal**.

> **Note:**
>
> - There is a one minute limitation on how frequently you can run scheduled integrations with an iCal expression. Anything below this limit is not supported.
> - You cannot use lookups in iCal expressions.

- Enter an iCal expression, and click **Validate Expression**. For example:
  - The following expression indicates that this integration runs each month on the 1st, 10th, and 15th days of the month at 5:15 AM, 10:15 AM, 3:15 PM, and 8:15 PM.

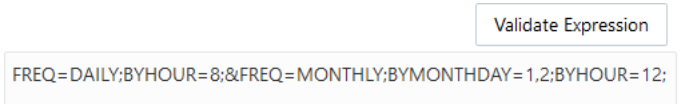


  - You can also define multiple schedule frequencies. The following schedule runs every day between the hours of 5:30 PM – 7:30 PM, and during these hours it executes every 10 minutes. This configuration requires three schedules separated by the `&` sign:

```
FREQ=DAILY;BYHOUR=17;BYMINUTE=30,40,50;BYSECOND=0;
&FREQ=DAILY;BYHOUR=18;BYMINUTE=0,10,20,30,40,50;BYSECOND=0;
&FREQ=DAILY;BYHOUR=19;BYMINUTE=0,10,20,30;BYSECOND=0;
```

  - The following schedule runs daily at 8 AM and also monthly at 12 PM on day 1 and day 2.

```
FREQ=DAILY;BYHOUR=8;&FREQ=MONTHLY;BYMONTHDAY=1,2;BYHOUR=12;
```



  If validation is successful, the following message is displayed at the top:

```
iCal expression is valid.
```

7. In the **This schedule is effective** section, click the link to the right of **From**.

   A menu is displayed for defining the start date of the schedule.

8. If you want to start the integration run when the schedule is activated, click **When schedule starts**.

9. If you want to explicitly set an integration run start date:

   a. Select **Modify start date**.

   b. Click the **Calendar** icon to select the month, year, and day and the hour, minute, and second at which to start the integration run.

c. Click **OK**.

10. In the **Until** section, click the link to the right.

    A menu is displayed for defining the expiration date.

11. If you want the schedule run to never expire, select **Never (repeat indefinitely)**.

12. If you want the integration run to have a fixed expiration date:

    a. Select **Choose expiry date**.

    b. Click the **Calendar** icon to select the month, year, and day and the hour, minute, and second at which to end the integration run.

    c. Click **OK**.

13. From the **Time zone** list, select a value. By default, this field shows the value you selected on the Preferences page.

14. Click **Save**, then click **Update & Save** when prompted.

    If successful, a message is displayed in the upper corner.

    ```
    Schedule Run name saved successfully
    ```

    If there are any errors, a validation message is displayed in the upper right corner that describes how to resolve the errors.

15. Click .

    The Schedule and Future Runs page is displayed.

    Details about the integration run schedule are displayed. You can click the icon to update the schedule definition.

    There are two sections on the Future Runs page:

    - Schedule section: This section is collapsed by default. Clicking the top bar (showing schedule name and buttons) expands the section and shows the schedule.
    - Future Runs table: This remains empty until the schedule is started. Once you start the schedule, you can see data.

16. If you have already activated the integration, select an option to run the integration:

    a. Select **Submit Now** from the menu.

    b. If you are a user with the ServiceAdministrator role, you can change the user with which to submit the schedule.

    Clicking **Submit Now** triggers the integration immediately. Clicking **Start Schedule** triggers the integration using the defined schedule. (with frequencies and other definitions).

    c. Select how to initiate an instance of the integration when prompted. Otherwise, click **Cancel**.

    - **Ad hoc request**: The instance is executed independently of the scheduled run for the integration (if any).
    - **As part of schedule**: The instance runs in sequence with the scheduled run and shares schedule parameters (if any).

    d. Click **Submit Now**.

    or

a. Select **Start Schedule** on the Schedule and Future Runs page to activate the integration schedule.

b. If you are a user with the ServiceAdministrator role, you can change the user with which to submit the schedule.

c. Click **Start** when prompted. Otherwise, click **Cancel**.

> **Note:**
>
> - If you place a file in an input directory and run an integration for the first time, the file is copied to the output directory. If you then run the integration a second time, the same file is not copied again to the output directory, even if you deleted the file from the output directory before rerunning the integration. This is by design. Second integration runs do not copy the same file. However, if a new file is placed in the input folder or the previously-copied file is updated with a newer version in the input directory, both are copied to the output directory.
> - If you create a new schedule to repeat every minute and launch the calendar to specify a start time, it shows the current time (for example, 12:41:16). If you update the start time to several minutes later (for example, 12:43:55), save, and start the scheduled integration run, the integration starts running at 12:43:16 instead of 12:43:55. The scheduler does not care about the seconds entered and runs the schedule every minute.

17. If you have not yet activated the integration, return to the Integrations page and click 
.

18. Run the scheduled integration as described in the previous steps.

Create and Update Parameters in Scheduled Integrations

You can create and update scalar-type scheduled parameters in scheduled integrations that determine how to batch and read data received from a source location. You then use these parameter values downstream in the integration. You create and assign values to these parameters in the Schedule Parameters page that is available in scheduled orchestrated integrations. You can also assign values to these parameters in assign actions. Up to five parameters are supported in the Schedule Parameters page.

Create Parameters

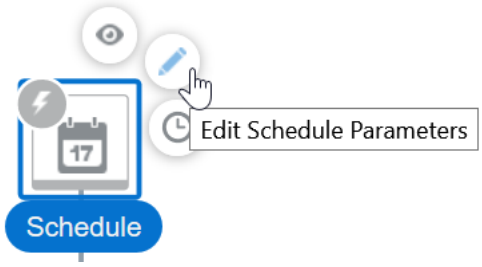
You can also create and assign scheduled parameters as business identifiers that enable you to track fields in messages during runtime. See Assign Business Identifiers.

Scheduled parameters are available across all scheduled runs of an integration and can facilitate processing of data from one run to the next. For example, when performing batch processing, a scheduled parameter can track the current position of batch data between runs. This value is available across scheduled runs and cannot exceed 256 characters.

To create parameters during design-time:

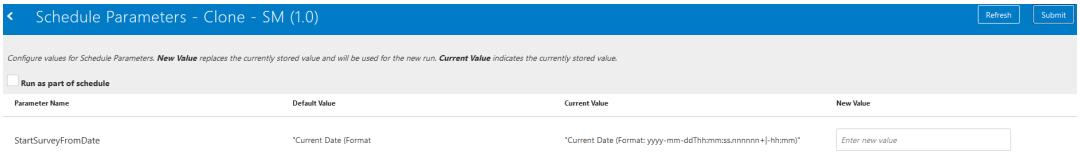
1. On the Integrations page, click **Create**.
2. Select the **Scheduled Orchestration** style.

3. Complete the fields of the dialog, and click **Create**.
   The integration canvas is displayed.
4. Click the **Schedule** icon, then select **Edit Schedule Parameters**.



.

5. In the **Parameter Name** column, click the **plus** icon.
6. Enter a name, an optional description, and a default value in the **Value** column. For this example, a parameter of `StartSurveyFromDate` and a value of `"Current Date (Format: yyyy-mm-ddThh:mm:ss.nnnnnn+|-hh:mm)"` are entered.



7. Click **Close** to save your changes and exit the page.
   If you enter a parameter without a value, a red warning icon is displayed on the **Schedule** icon. If you place your cursor over the icon, a message indicates that one or more parameters are missing a value.
8. Drag an **Assign** action into the integration, enter a name, and then click **OK**.
9. In the **Name** column, click **+**, then select the parameter from the dropdown list that you created without a value in the Schedule Parameters page.
10. Click the
    
    icon to invoke the Expression Builder.
11. Create an expression, then click **Close**.
12. In the Assign page, click **Close**.
    You are returned to the integration canvas.
13. Place your cursor over the **Schedule** icon, and note that the message about the missing parameter value is gone.

> **Note:**
>
> If you create a for-each action without selecting the **Process items in parallel** option and then add an assign action within the for-each loop, parameters defined in scheduled integrations are visible for selection in the assign action. If you later edit the for-each action and select the **Process items in parallel** option, a red fault icon is displayed on the impacted assign action indicating that a scheduled parameter is being used. You must correct this error for your integration to be valid. You can disable the **Process items in parallel** option or use the **Reposition** button to move the assign action outside of the for-each loop.

Update Parameters

1. Go to the Integrations page.
2. Go to the row of the integration on which the scheduled run is defined.
3. Select **Schedule** from the  menu.
   The Schedule and Future Runs page is displayed.
4. Click the **Update Schedule Parameters** link in the  menu.
5. Edit parameter values in the **New Value** column, and click **Update**.

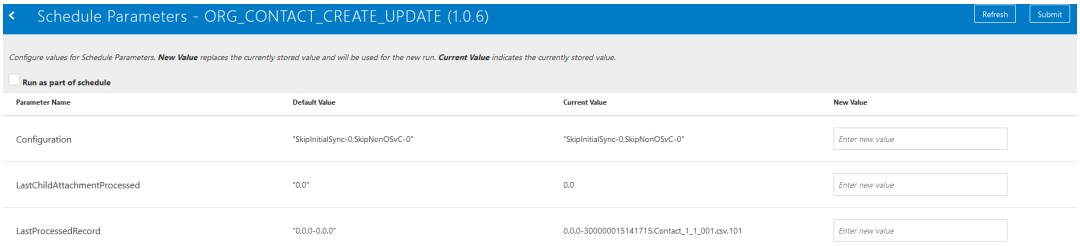
Override Schedule Parameter Values

You can override schedule parameter values set at design-time when submitting an ad-hoc schedule request or starting a schedule in an activated integration. This feature enables you to provide parameter values while invoking an integration without deactivating it.

To override schedule parameter values:

1. Access the dialog to override parameter values in either of two ways. If the scheduled integration has defined parameters, the Schedule Parameters dialog is displayed. Otherwise, the dialog is not displayed and the scheduled integration starts.
   a. If submitting an ad-hoc schedule integration, select **Submit Now** from one of the following locations:
      - Select **Submit Now** from the  menu on the Schedule and Future Runs page.
      - Click the  icon on the Monitor Integrations page.
   b. If scheduling an integration, select **Start Schedule** on the Schedule and Future Runs page.

   The Schedule Parameters dialog is displayed.



2. Make updates to any necessary parameters.

   The following fields are displayed:

   - **Parameter Name**: Displays the parameter name you provided during design-time.
   - **Default Value**: Displays the parameter value you assigned during design-time.
   - **Current Value**: Displays the value most recently used at runtime.
   - **New Value**: Optionally enter new values for the schedule parameters. This value overrides the current and default values. If this field remains empty, the current value is used. If there is no new or current value, the default value defined during design-time is used.

3. In the **New Value** row for the specific parameter, enter a value.

   If the integration updates these schedule parameter values using an assign action, the updated value is saved and becomes the current value for the next schedule run.

4. Click the button that is displayed to save your updates. The name of the button is based on the type of schedule parameter you are updating:

   a. If starting a schedule, click **Start Schedule**.

   b. If submitting an ad-hoc scheduled integration, click **Submit Now**.

   The values you entered are used during execution of the schedule integration.

Start and Pause an Integration Schedule

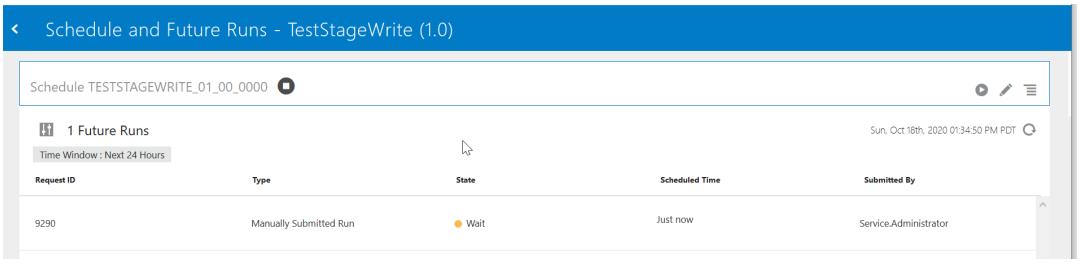
After you define a schedule, you must activate it. You can also pause (deactivate) a schedule, as needed. You can also submit schedules to run immediately.

To start an integration schedule:

1. Go to the Integrations page.
2. Go to the row of the integration on which the scheduled run is defined.
3. Select **Schedule** from the  menu.

   The Schedule and Future Runs page is displayed.
4. Start the scheduled integration in either of the following ways.

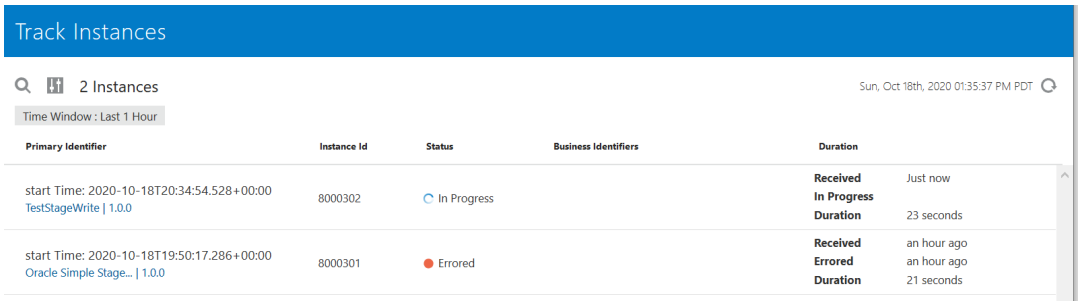
Method	Steps
To start an integration based on a schedule. **a.** Click the  icon. **b.** If you are a user with the ServiceAdministrator role, you can change the user with which to submit the schedule. **c.** Click **Confirm** to start the schedule. If you have defined scheduled parameters on this integration, the Schedule Parameters page is displayed. **d.** Make any necessary parameter value updates, and click **Start Schedule**. The following message is displayed in the banner. `Pause the schedule when you need to temporarily prevent this integration from running.` The  button is changed to  (stop schedule button). **Note:** If you want to pause the schedule run, click  in the upper right corner, then click **Confirm** when prompted.	To start a schedule immediately: **a.** Select **Submit Now** from the  menu. **b.** If you are a user with the ServiceAdministrator role, you can change the user with which to submit the schedule. The following message is displayed in the banner. `Submit Now request to run integration integration_name (version_number) was submitted successfully with instance id number.`



5. Click the instance ID in the banner or select **Track Instances** from the  menu to track scheduled integration progress on the Track Instances page.

   When you click **Submit Now** or **Start Schedule**, an instance status and instance ID are created immediately and are visible on the Track Instances page. The following status values are displayed during the course of the scheduled integration lifecycle:

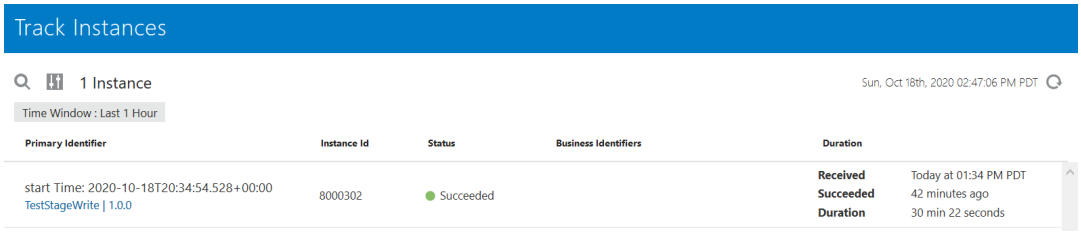
- Waiting (Typically the state of an instance when it is scheduled to execute sometime in the future.)
- Paused (The schedule has been paused, meaning the instance is also paused. Unpausing the schedule moves the instance back to the waiting state.)
- In progress
- Succeeded
- Errored
- Aborted



If necessary, scheduled integration instances in the waiting or paused state can be canceled by selecting the

icon. This action moves the scheduled integration instance to the aborted state. If this is a regularly scheduled integration (for example, it is scheduled daily at the same time), another instance gets created immediately for the next day's schedule and moved to the waiting state.

If the scheduled integration run is successful, the status value is updated to **Succeeded**.



6. Click

to view the following schedule milestones in the activity stream.

- Schedule request submitted: Indicates the time at which the request to trigger the scheduled integration was submitted.
- Schedule request started running: Indicates the time at which the submitted request started executing.
- Schedule paused: Indicates the time at which a schedule was paused.
- Schedule resumed: Indicates the time at which the schedule was resumed (from a paused state).

7. Select **Schedule** from the menu if you want to return to the Schedule and Future Runs page.

Change the User Submitting an Integration Schedule

You can designate a specific user to control the schedule of an integration in the absence of the actual submitter. This allows integrations to be scheduled using a service account that does not go away if a user leaves the company or moves to another department. Only a user with the ServiceAdministrator role can change the submitter of a schedule.

1. Go to the Schedule and Future Runs page. See Define the Integration Schedule.
2. Select **Update Submitter** from the menu to update the user that can submit the schedule.

   The Update Submitter dialog is displayed.
3. Change the submitter of the schedule in the dropdown list.

   This update is per a specific integration schedule, and not globally for all integration schedules that can be submitted by this user. You can globally change the submitter of integration schedules under the **Settings** > **Integrations** tab. See Globally Change the Submitter of Integration Schedules.

   Details about the user who submitted the schedule for the integration are visible in the design time audit logs on the Dashboards page. For example:

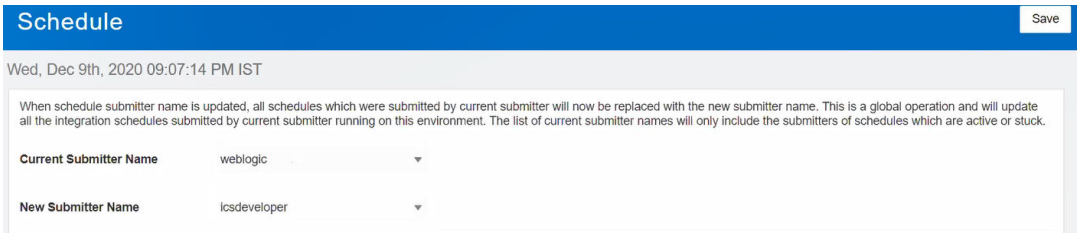
   ```
   User icsadmin triggered integration integration_name as user icsdeveloper
   ```

   Child (co-located) integrations invoked from a parent scheduled integration inherit the same user that submitted the schedule. See Invoke a Co-located Integration from a Parent Integration.

Globally Change the Submitter of Integration Schedules

You can globally change all schedules submitted by the current user to another user. The list of current submitter names only include submitters of those schedules that are currently active or stuck. Only a user with the ServiceAdministrator role can perform this task.

1. In the left navigation pane, click **Home** > **Settings** > **Integrations** > **Schedule**.
2. From the **Current Submitter Name** list, select the submitter to replace.
3. From the **New Submitter Name** list, select the new submitter.



For this example, any schedules that the old **weblogic** user submitted can now be submitted by the new **icsdeveloper** user.

4. Click **Save**.

View Past and Present Integration Runs

You can view the status of past and present scheduled integration runs.

To view past and present integration runs:

1. Go to the Integrations page.
2. Go to the row of the integration on which the scheduled run is defined.
3. Select **Schedule** from the  menu.

   The Schedule and Future Runs page is displayed.
4. Select **Track Instances** from the  menu.

   The Track Instances page is displayed. By default, all completed integration runs are displayed.
5. Filter the display of integration runs under
   
   .
6. Click the file name of the instance.

   A graphical view of the integration flow is displayed.

   For example, if the resubmission resulted in a failure, details are displayed.

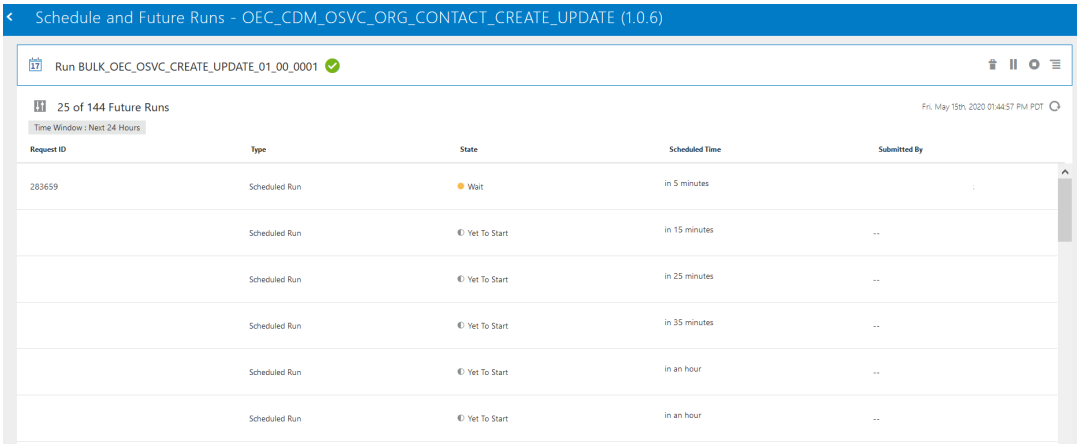
View Future Runs

You can view the status of future scheduled integration runs.

To view future runs:

1. Go to the Integrations page.
2. Find the integration on which the scheduled run is defined.
3. Select **Schedule** from the  menu.

   The Schedule and Future Runs page is displayed.
   Details about any future runs that have been scheduled are displayed. You can filter to show all runs, only automatically scheduled runs, or only manually submitted runs. The page only displays manually submitted runs that have a state of blocked, wait, ready, paused, or canceling. The time zone in which the schedule was created is also displayed.



> **Note:**
>
> If you select a different time zone in another browser session, the **Schedule Time zone** field continues to show the time zone in which the schedule was created. All future runs execute as per this time zone, and not the time zone you specified in another browser session.

See Run the File Transfer Sample for a sample of a scheduled orchestrated integration.

Edit an Integration Schedule

You can edit a schedule for an integration run.

To edit an integration schedule:

1. Go to the Integrations page.
2. Go to the row of the integration on which the scheduled run is defined.
3. Select **Schedule** from the
   
   menu.

   The Schedule and Future Runs page is displayed.
4. On the far right, click
   
   .
5. Edit the schedule. See Define the Integration Schedule.
6. Click **Save**.

Delete an Integration Schedule

You can delete a schedule for an integration run.

To delete an integration schedule:

1. Go to the Integrations page.

2. Go to the row of the integration on which the scheduler run is defined.
3. Select **Schedule** from the  menu.

   The Schedule and Future Runs page is displayed.
4. Find the schedule in the list that you want to delete.
5. Click  .
6. Click **Confirm** when prompted.

Unlock an Integration Schedule

When a schedule is in edit mode and the browser crashes, the schedule becomes locked, which prevents it from being edited. You can unlock an integration schedule to resume editing.

To unlock the integration schedule:

1. Log in as a user with the Administrators role.
2. In the left navigation pane, click **Home** > **Integrations** > **Integrations**.
3. Select **Schedule** from the  menu.

   The Schedule and Future Runs page is displayed.
4. Select **Unlock Schedule** from the  menu.
5. Click **Unlock** when prompted. Note that unlocking an integration schedule may cause data loss.

Convert a Scheduled Integration to a REST Adapter-Triggered Orchestration Integration

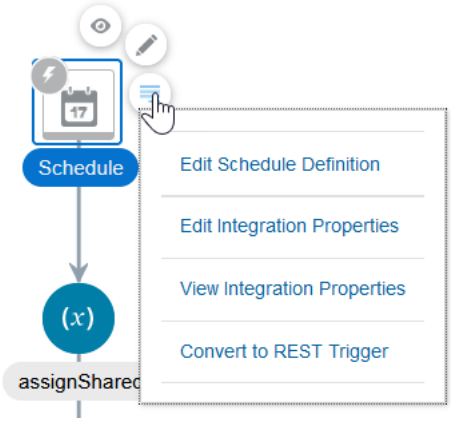
You can convert a scheduled integration to a REST Adapter-triggered, app driven orchestration integration. This eliminates the need to recreate the integration if you need to change the integration style from scheduled to app driven orchestration.

Note the following restrictions:

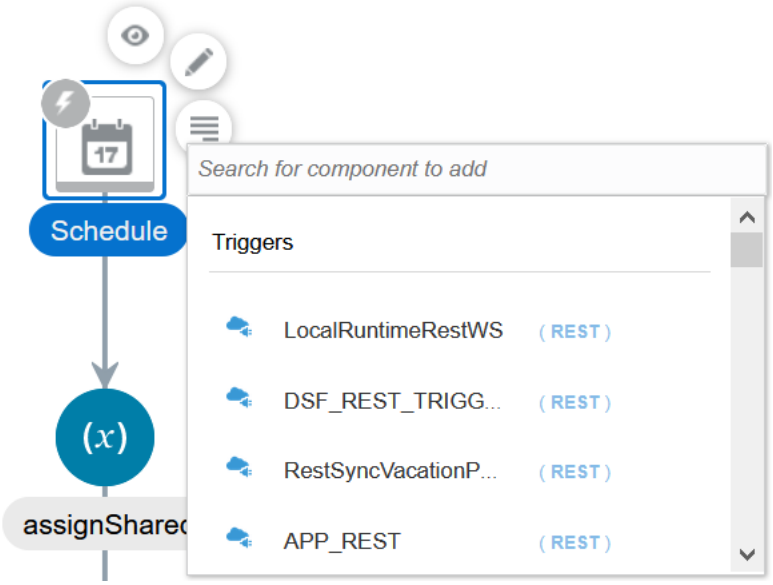
- You cannot convert a scheduled integration to a REST Adapter-triggered integration with multiple verbs.
- You cannot convert a scheduled integration with assignments that have the fault data type.

1. In the left navigation pane, click **Home** > **Integrations** > **Integrations**.
2. Open the scheduled integration you want to convert.

   The integration canvas is displayed.
3. Right-click the schedule icon and select **Convert to REST Trigger**.



A list of available REST Adapter triggers is displayed.



4. Select the REST Adapter to use as the trigger connection in this integration.

   This invokes the Adapter Endpoint Configuration Wizard.

5. Configure the REST Adapter in the wizard.

> **Note:**
>
> Do *not* select the **Select to configure multiple resources or verbs** checkbox on the Basic Info page. This feature is not supported when converting scheduled integrations to app driven orchestration integrations.

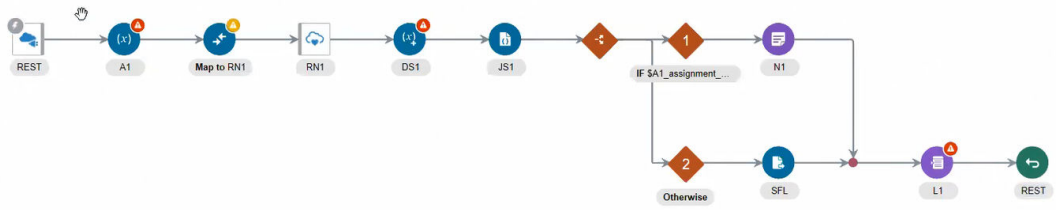
6. When wizard configuration is complete, click **Done**.

A message is displayed.

```
Clicking done will perform conversion to REST trigger
and will clear tracking information, delete any assignment variables
with fault data types and update any impacted actions. Errors and
warnings may occur which will require manual correction. Are you
sure you want to continue?
```

7. Review the message, then click **Convert**.

   A conversion progress bar is displayed. When complete, the converted integration is displayed in the canvas with the REST Adapter trigger connection.



8. Investigate any errors and manually correct the impacted actions or invokes. For example, any scheduled parameters in the scheduled integration are deleted during conversion to an app driven orchestrated integration. This results in errors in any actions that reference those schedule parameters (for example, an assign or logger action).

Delete Runs

You can gracefully delete scheduled or ad-hoc runs in a number of states. Only a user with the administrator role can perform this activity. Any running instances triggered by these scheduled and ad-hoc runs are also terminated when you delete the run.

You can delete the following states for both scheduled and ad-hoc runs:

- Wait
- Ready
- Running
- Completed
- Blocked
- Hold
- Paused
- Schedule Ended
- Error Auto Retry
- Error Manual Recovery

To delete a run:

1. Go to the Integrations page.
2. Go to the row of the integration on which the scheduler run is defined.

3. Select **Schedule** from the

menu.

   The Schedule and Future Runs page is displayed. The state of each run is displayed.
4. Select the scheduled run to delete.
5. Click

, then click **Yes** when prompted to confirm your selection.

   The scheduled run is removed from the Schedule and Future Runs page.

Monitor Integration Runs

You can pause and resume scheduled runs for an integration from the Monitoring page.

To monitor integration runs:

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

   For scheduled orchestrated integrations, the

icon is displayed.
2. If you want to resubmit a run of a scheduled integration, click

.
3. If you want to pause the scheduled runs, select **Pause Schedule** from the

menu.

Understand the Behavior of Scheduled Integrations When Using the Local Time Zone

You should understand how scheduled integrations are impacted by use of the local time zone. This section provides several use cases.

Daylight Savings Time Begins

Assume daylight savings begins in your environment on March 1 (time is moved forward from 1 AM to 2 AM) and your integration is scheduled to run at 1:30 AM. In this scenario, the time range between 1 AM and 1:59 AM does *not* exist and the scheduled integration does *not* run on March 1 at 1:30 AM.

Daylight Savings Time Ends

Assume daylight savings time ends in your environment on November 1 (the time is rolled back from 2 AM to 1 AM) and your integration is scheduled to run at 1 AM. The scheduled integration runs as expected at 1 AM. When 2 AM arrives and the time is rolled back to 1 AM, the scheduled integration does *not* run again.

Create Ad-Hoc Integration Runs Using the REST API

You can create ad-hoc runs on integrations on which a schedule has been defined from the REST API. This is useful for when you want to test a scheduled integration.

To create ad-hoc integration runs:

1. In the navigation pane, click **Home** > **Integrations** > **Integrations**.
2. Go to the row of the integration on which a schedule is defined (indicated by the **calendar** icon).
3. Click the  icon.
4. Copy the URL from the **Submit Now Link** section into a tool to test REST APIs (for example, SOAP UI, but you can use any REST invocation tool or script). You can also use CURL commands from a terminal window to invoke the REST APIs.
5. Invoke the integration run.
6. Return to the Integrations page, and select **Schedule** from the  list.

   The Schedule and Future Runs page is displayed.
7. Click the **Track Instances** link in the  menu.

   Details about the ah-hoc integration run are displayed.

Activate Older, Imported Scheduled Integrations Using Operations Other Than Read Multiple Files

If an older, scheduled integration in which the first FTP Adapter uses an operation other than Read Multiple Files is imported into Oracle Integration, it cannot be activated.

To work around this issue, perform the following steps:

1. Edit the integration.
2. Remove **Filename** as a business identifier tracking variable.
3. Save and activate the integration.

6

Create Basic Routing Integrations and Integrations to Publish and Subscribe to Oracle Integration

You can create basic routing integrations and integrations to publish and subscribe to Oracle Integration.

Topics:

Create Basic Routing Integrations

You create an integration that provides a template with empty trigger and invoke connections in which to add your own adapters. You can also create a single routing expression and request and response enrichments, as needed. You cannot create multiple routing expressions. If your integration requires this feature, create an orchestrated integration.

Topics:

> **Note:**
>
> The basic routing integration style has been deprecated. Oracle recommends that you use the app driven orchestration integration style, which provides more flexibility. You can migrate basic routing integrations to app driven orchestration integrations. See Convert a Basic Routing Integration to an App Driven Orchestration Integration.

Create a Basic Routing Integration

This section describes how to create a basic routing integration.

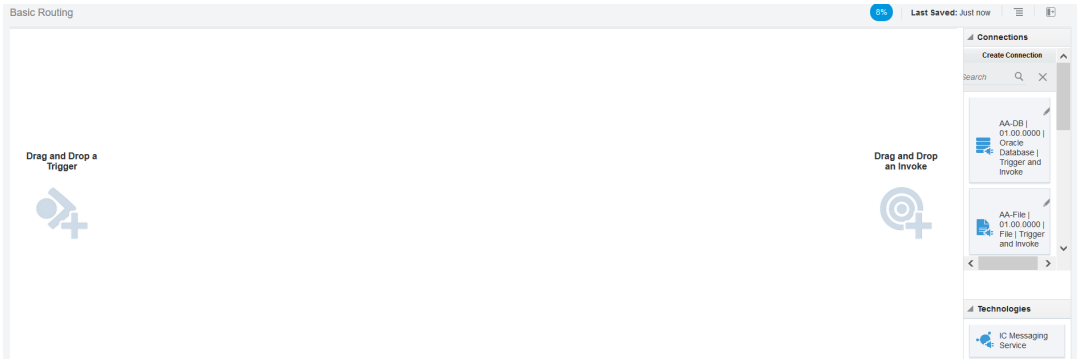
1. Follow the steps in Create Integrations to create a basic routing integration.
   An integration canvas with empty trigger and invoke connections is displayed.

Add a Trigger (Source) Connection

The trigger (source) connection sends requests to Oracle Integration. The information required to connect to the application is already defined in the connection. However, you still must specify certain information, such as the business object and operation to use for the request and how to process the incoming data.

To add a trigger connection:

- In the integration canvas, drag a connection from the **Connections** or **Technologies** panel on the right to the **Trigger** (source) area on the canvas.



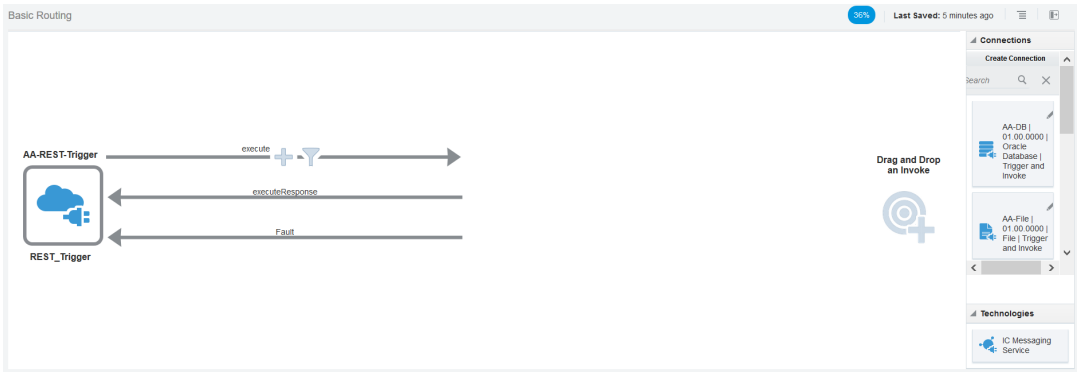
The Adapter Endpoint Configuration Wizard for the selected connection is displayed. The pages in the wizard that appear are based on the adapter you selected. See Understand Trigger and Invoke Connections.

Add an Invoke (Target) Connection

Oracle Integration sends requests or information to the invoke (target) connection. The information required to connect to the application is already defined in the connection. However, you still must specify certain information, such as the business object and operation to use for the request and how to process the data.

To add an invoke (target) connection:

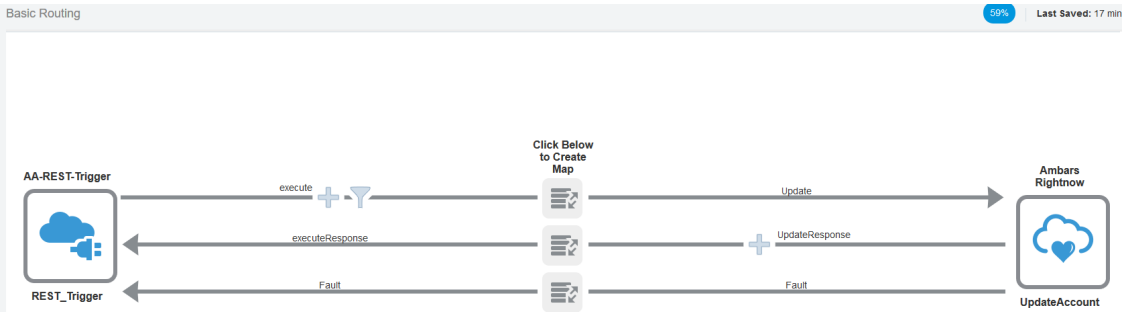
1. In the Integration canvas, drag a connection from the **Connections** or **Technologies** panel on the right to the **Invoke** (target) area on the canvas.



The Adapter Endpoint Configuration Wizard for the selected connection is displayed. The pages in the wizard that appear are based on the adapter you selected. See Understand Trigger and Invoke Connections.

2. After you configure the connection, the Summary page appears.
3. Click **Done**, then click **Save**.

The connection information appears on the canvas, along with arrows depicting the configured operations. Because of space limitations on the canvas, names of connections that are more than 15 characters are truncated and ellipses are added. If you hover over a name, the complete name is displayed in a tool tip.



Add Request and Response Enrichments

When you create an integration, you also have the option of adding both request and response message enrichment points to the overall integration flow. Enrichments participate in the overall integration flow and can be used in the request and/or response payloads between the trigger and invoke.

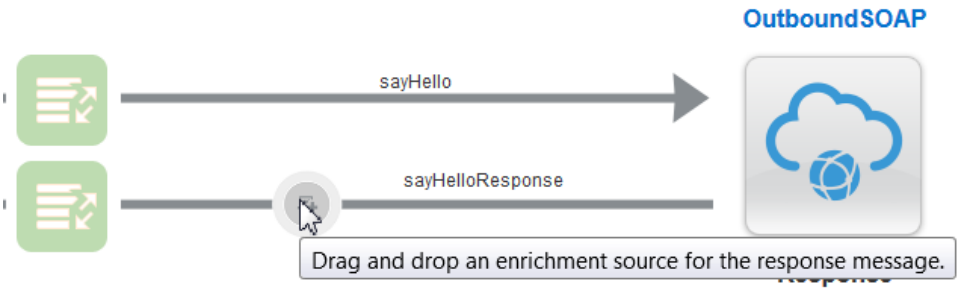
To add request and response enrichments:

1. Design an integration with trigger and invoke connections and request and response mappings. For this example, the integration looks as follows when complete. Note the two enrichment point circles in the design; one appears on the inbound (request) side and the other appears on the outbound (response) side.

The request and response mappings for this example are as follows:

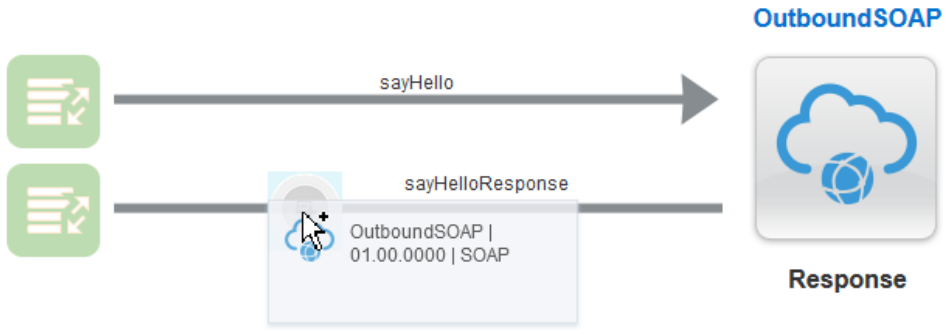
Mapping	Source	Target
Request	HelloRequest/FirstName	sayHello/name
Response	sayHelloResponse/ sayHelloReturn	HelloResponse/Greeting

You are now ready to add enrichments to the integration. For this example, a response message enrichment is added to the **Drag and drop an enrichment source for the response message** area. You can also add request message enrichments on the request (inbound) side.



2. From the **Connections** panel on the right, drag an adapter to the enrichment area on the response message shown below.

   For this example, a SOAP Adapter is dragged to the **Drag and drop an enrichment source for the response message** area. This action invokes the wizard for configuring the SOAP Adapter.



3. Complete the pages of the wizard to configure the SOAP Adapter, then click **Done**. For this configuration, a different operation for selecting timestamp details is chosen.

   You are prompted with a dialog to delete any impacted response mappings that you previously configured for the response mapper. The response mapper requires updates because of the enrichment response adapter configuration you just performed.

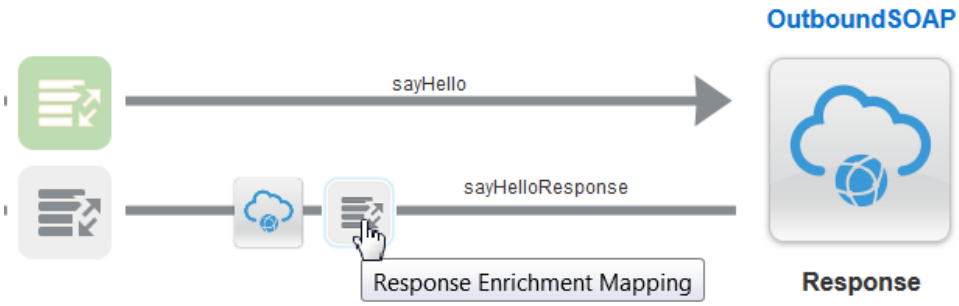
4. Click **Yes**. You recreate the response mappings later in these steps.

5. Click **Save**.

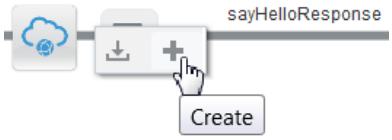
   A SOAP Adapter icon and response enrichment mapper are added to the response side of the integration. Note that because you deleted the response mappings in the previous

step, that icon is no longer shaded in green. This indicates that the response mapper requires configuration.

6. Click the **Response Enrichment Mapping** icon between the trigger and invoke.



7. Click the **Create** icon that is displayed. This invokes the mapper.



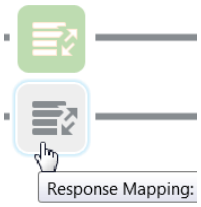
8. Map source elements to target elements to include a timestamp with the response, then click **Save** when complete.

   The response enrichment mappings are as follows:

Mapping	Source	Target
Response Enrichment	sayHelloResponse/ sayHelloReturn	visitTimestampReq > reqMsg

   The **Response Mapping** icon is displayed in green, indicating that it has been configured.

9. Click the **Response Mapping** icon to invoke the mapper again. This mapper requires updates because of the enrichment response mapping you performed.



10. Remap the source elements to target elements in the response mapper.

    The response mappings are updated. Note that a different source is now mapped to the original target of HelloResponse/Greeting.

Mapping	Source	Target
Response	$ResponseEnrichmentApplicationObject > visitTimestampResp > respMsg	HelloResponse/Greeting

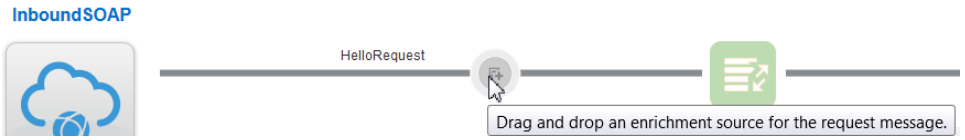
The **Response Enrichment Mapping** icon is displayed in green, indicating that it has been reconfigured.

11. Click **Close**, then click **Apply** when complete.

    The integration with response enrichments added to the invoke (target) area looks as follows:

12. Click **Save**, then click **Close** when complete.

    You are ready to activate the integration. While not demonstrated in this example, you can also configure the enrichment area on the request message shown below by dragging and dropping an adapter to the **Drag and drop an enrichment source for the request message** area. This invokes the adapter configuration wizard.



You can also update existing enrichments at a later time, such as the objects selected in the adapter configuration wizard and the enrichment mappings.
See About Oracle Integration Enrichments.

Delete Request and Response Enrichments

You can delete the request and response message enrichment point mappings added to an integration. After deleting the enrichment point mappings, the integration is returned to its original pre-enrichment state.

To delete request and response enrichments:

1. On the Integrations page, select the integration. The integration must not be active.
2. Click the enrichment area on the request message or response message to delete.
3. Select the **Delete** icon that is displayed.

   This deletes the mappings.
4. Click **Yes** when prompted to confirm.

   Click **Save**, then click **Close**.

See About Oracle Integration Enrichments.

Create Routing Paths for Two Different Invoke Endpoints in Integrations

You can create an integration in which you define routing paths for two different invoke endpoints. During runtime, the expression filtering logic for the routing paths is evaluated and, based on the results, the path to one of the invoke endpoints is taken. If the filtering logic for neither routing path is satisfied, then neither invoke endpoint is contacted.

The expression logic works as follows:

- You define an expression filter on the first (upper) invoke endpoint.
- You define either an ELSE condition or an expression filter on the second (lower) invoke endpoint.

During runtime, if the expression filtering logic for the first (upper) invoke endpoint evaluates to true, then the path to that invoke endpoint is taken. If the expression evaluates to false, then that invoke endpoint is skipped, and the path to the second (lower) invoke endpoint is taken through either an ELSE condition or an expression filter.

In addition to creating routing paths, you also define request and response (and optionally, enrichment) mappings on both invoke endpoints.

To create routing paths for two different invoke endpoints in integrations:

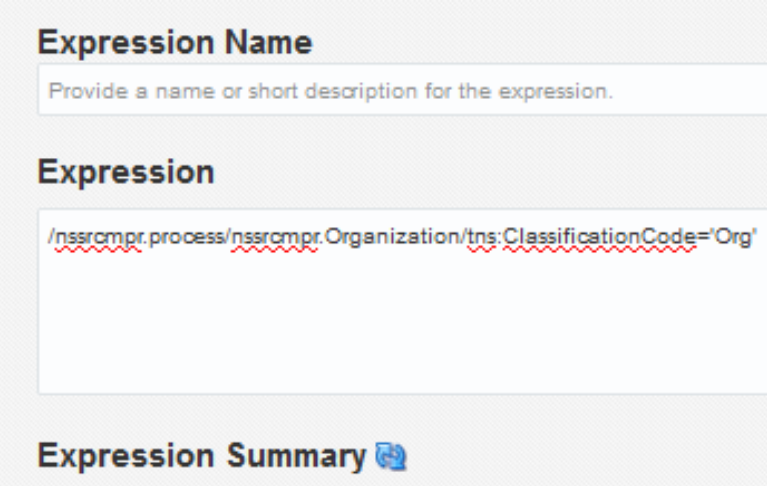
1. On the Integrations page, select the integration in which to define a routing filter. Ensure that the integration is fully defined with trigger and invoke connections, business identifier tracking, and mappings.
2. Click the **Filter** icon on the trigger side of the integration to create a filtering expression. Routing is created after any defined request enrichment and before the initial request mapping.

3. Click the **Routing** icon in the menu that is displayed.

   The Expression Builder is displayed for building routing expressions. The Expression Builder supports multiple source structures. You can create OR expressions using both source structures. You can also name expressions and calculate expression summaries with the **Expression Summary** icon. Elements and attributes with and without namespace prefixes are also supported.
   You can filter the display of source structures by clicking the **Filter** link. This enables you to filter on whether or not fields are used and on the type of field (required fields, custom fields, or all fields). You can also select to filter both required and custom fields together.
4. Drag an element from the **Source** area to the **Expression** field.
5. Define a value.

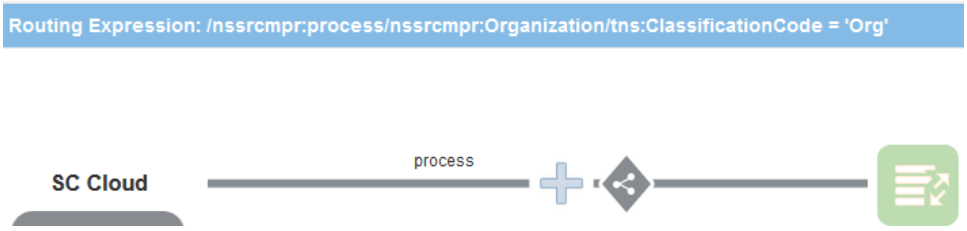
   For this example, the **ClassificationCode** element is defined as equal to `Org`. This means that `Org` is retrieved when this expression evaluates to true.
6. If you want to calculate the expression, click the **Expression Summary** icon. This shows the summary of the expression and defines a more user-friendly, readable version of the expression you just created.

7. If that name is not sufficiently user-friendly, copy and paste the expression to the **Expression Name** field for additional editing.



8. Click **Close** to save your changes.

   The defined expression is displayed above the integration. The **Filter** icon has now changed to indicate that an expression is defined.



9. On the right side of the integration, click the **Routing Drawer** icon to display a graphical routing diagram with two potential paths. The first route that you just defined (the upper trigger and invoke) shows the defined expression above the line. The second route (the lower trigger and invoke) is displayed as a dotted line because it is not yet defined.

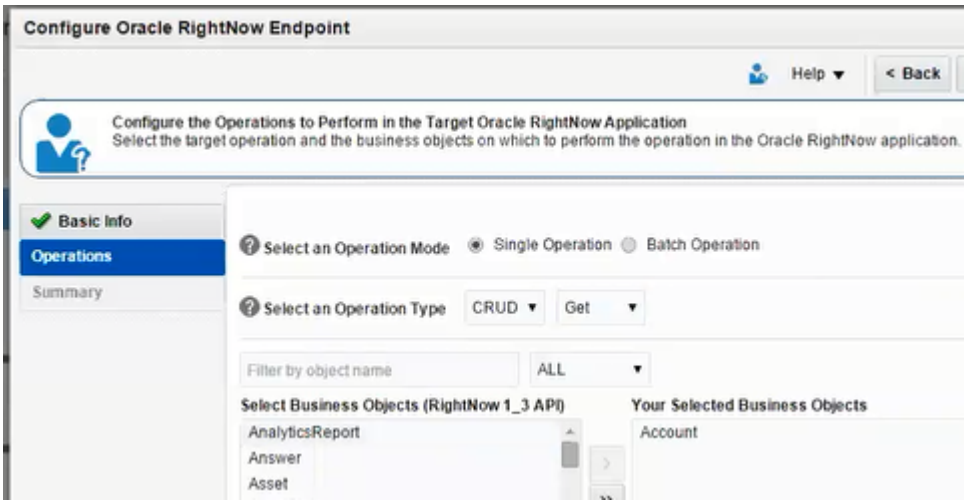
   You can activate the integration now if additional filtering is not required or define an additional routing filter. For this example, a second route is defined.

10. Click the **bull's eye** icon in the lower trigger icon to define routing on the second trigger and invoke route.

This refreshes the integration to display the lower trigger and invoke route in the integration. The trigger side remains as defined for the first route, but the invoke route is undefined.

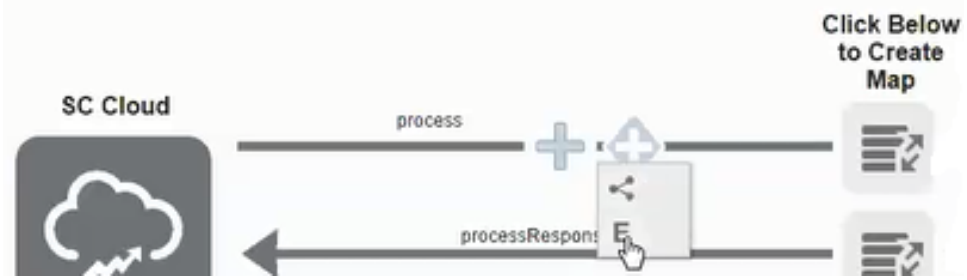
11. Click **Show Palette** to display the list of available connections and technologies.
12. Drag an adapter to the invoke (target) area of the integration (for this example, an Oracle RightNow adapter is added).

    The Adapter Configuration Wizard is invoked.
13. Configure the pages of the wizard for the Oracle RightNow adapter. For this example, the **Get** operation and **Account** business object are selected on the Operations page.



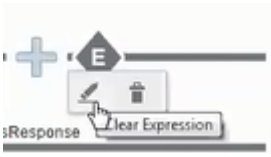
The integration is now defined for the second invoke. You now need to create a filtering expression for the second invoke.

14. Click the **Filter** icon to create a filtering expression.
15. If no additional expression is required, click the **E** icon (to create an ELSE condition).



This defines an ELSE condition for the second trigger and invoke. The ELSE condition is taken if the first route evaluates to false (that is **ClassificationCode** does not equal **Org**). You can toggle back and forth between the two trigger routes by clicking the adapter icon on the individual line. The line in blue is the currently visible invoke in the integration.

16. If you want to define your own expression filter for the second route instead of using the ELSE condition, perform the following steps:
    a. Click the **Filter** icon.
    b. Select **Clear Expression** to remove the ELSE condition.



    c. Click **Yes** when prompted to confirm.
    d. Click the **Filter** icon again and select the **Edit** icon to invoke the Expression Builder as you did in Step 3.
    e. Define an expression.
    f. Click **Close** to save your changes.

    Request and response mappings must now be defined.

17. Click the **Request Mapper** icon to define the mapping.

    For this example, the following mapping is defined.

Source	Target
process > Organization > Organizationid	Get > Account > ID > id

18. Click the **Response Mapper** icon to define the mapping.

    For this example, the following mapping is defined.

Source	Target
process > GetResponse > Account > ID > LookupName	processResponse > Organization > Name

    Integration design in now 100% complete.

19. Activate the integration.

Create Routing Expression Logic in Both Expression Mode and Condition Mode

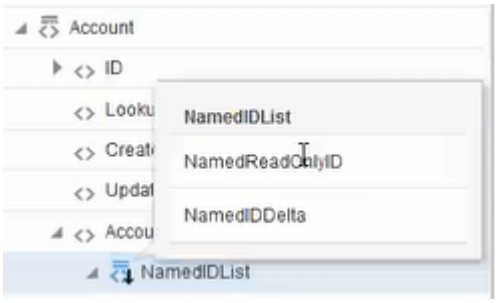
You can create XPath expressions for routing conditions in two different user interface modes:

- Expression mode: This mode provides an interface for creating and viewing the entire XPath expression.

- Condition mode: This mode provides an easier-to-read interface to create and view XPath condition expressions. This mode is useful for business analysts who may be less experienced with XPath expressions.

You can toggle between expression mode and condition mode when creating and viewing your expressions. Elements and attributes for which mapping is required are identified by a blue asterisk (*) to the left of their names. You can also place your cursor over elements and attributes to display specific schema details such as the data type, if mapping is required, and so on. When creating an expression, note the following functionality in the tree:

- Three levels of elements are loaded by default in the tree in the **Source** area. When you reach the third level, a **Load more** link is displayed. Click this link to display all the direct children of that element. Only base types are loaded automatically. To load the extended types of the base type, click the base type, which is identified by a unique icon. This invokes a menu of extended types that you can select to load one by one into the tree.



- Elements in the tree in the **Source** area that you have already dragged to an expression are identified by green checkboxes. These elements are displayed even if they are deeper than three levels in the tree.
- You can search for an element that is not yet loaded in the tree by entering the name in the **Find** field and clicking the **Search** icon. This action loads that specific element into the tree.

This section provides an example of building an expression using both modes.

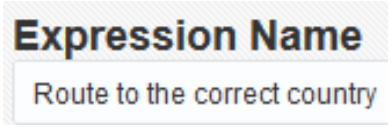
To create routing expressions in both expression mode and condition mode:

1. Click the **Filter** icon on the source side of an integration to create a filtering expression.

2. Click the **Routing** icon in the menu that is displayed.

   The Expression Builder is displayed for building routing expressions. Expression mode is the default mode.

3. In the field immediately below **Expression Name**, optionally enter a short description about the expression you want to build.

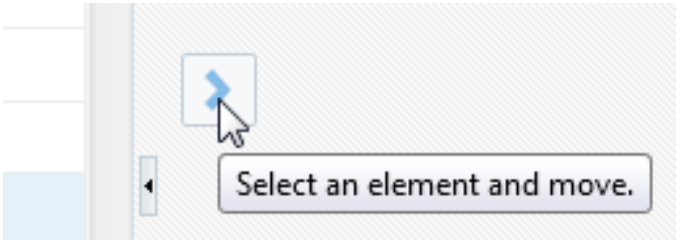


4. Add an element from the **Source** area on the left side to the expression field immediately below the short description field. If needed, you can also add functions from the **Components** section.

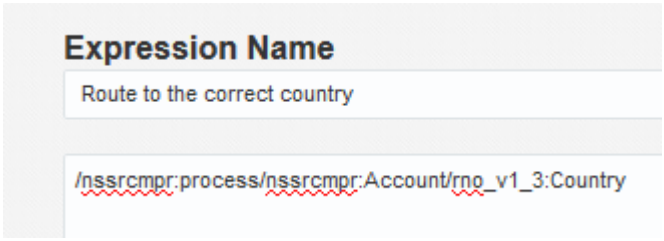
   There are two ways to add an element to the expression field:

   a. Drag the element from the **Source** area.

   b. Select the row of the element in the **Source** area, then click the **Move** icon in the middle of the page to move the element.

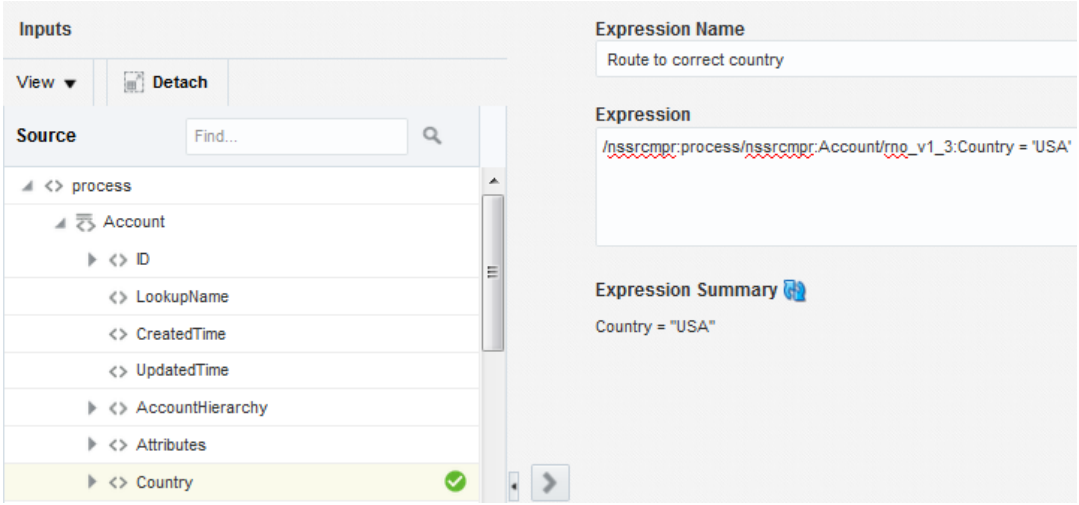


   The expression for the selected element is displayed in the expression field (for this example, the expression for the **Country** element was added). The selected element is identified by green checkbox in the **Source** area.



5. To the right of the added expression, define an operator and a value within single or double quotes (for this example, `= "USA"` is defined).

6. Click the **Expression Summary** icon to view a simplified, user-friendly version of the expression. Easy-to-read output is displayed.



> **Note:**
>
> - To add additional elements to the expression, you can place your cursor in the exact location of the expression, select the row of an element in the **Source** area, and click the **Move** icon. These actions add that element to the exact location of your cursor.
> - You can drag an element to the exact location of your cursor in the expression, and the expression of the element is added to the cursor location, and not the location in which you drop the element.
> - You can drag an element on top of an existing expression element to replace it.

7. In the upper right corner, click **Condition Mode** to view the expression you created in condition mode. Condition mode provides an easy-to-read interface for creating and viewing your expressions.

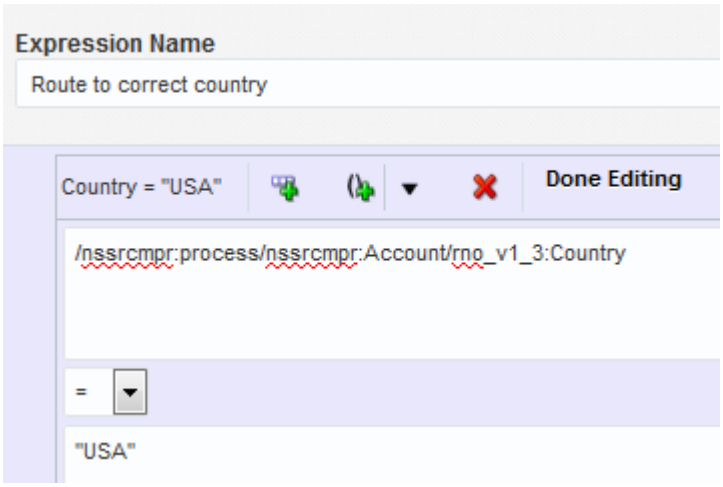
   Note the following details about accessing condition mode:

   - Condition mode can only be accessed if the expression field is empty or completely defined with an expression that returns true or false. If you only partially define an expression (for example, you drag an element to the expression field, but forget to define expression logic and a value such as `= "USA"`), you receive an error saying that you must provide a valid condition to access condition mode.
   - The **Condition Mode** button toggles to **Expression Mode**.

   > **Note:**
   >
   > At any time, you can click **Expression Mode** to view the entire XPath expression.

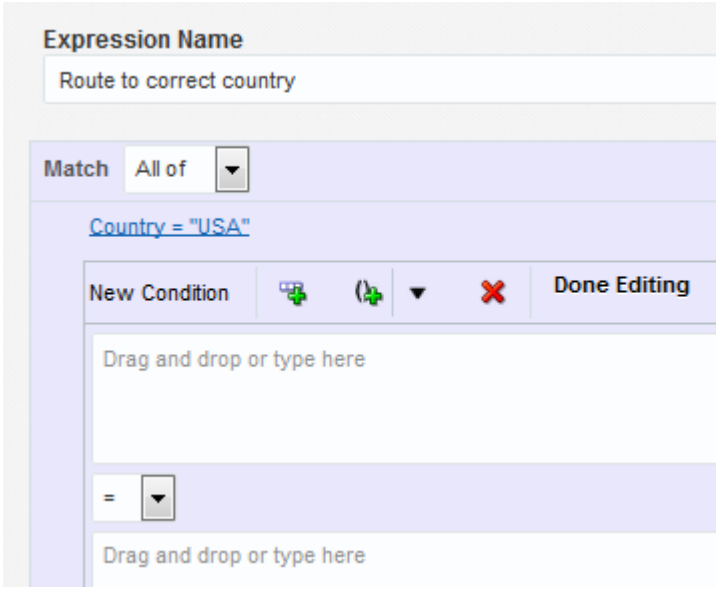
8. Click the expression.

This refreshes the page to display icons for adding additional conditions and conditions groups. Groups enable you to combine multiple conditions into a single logical expression.



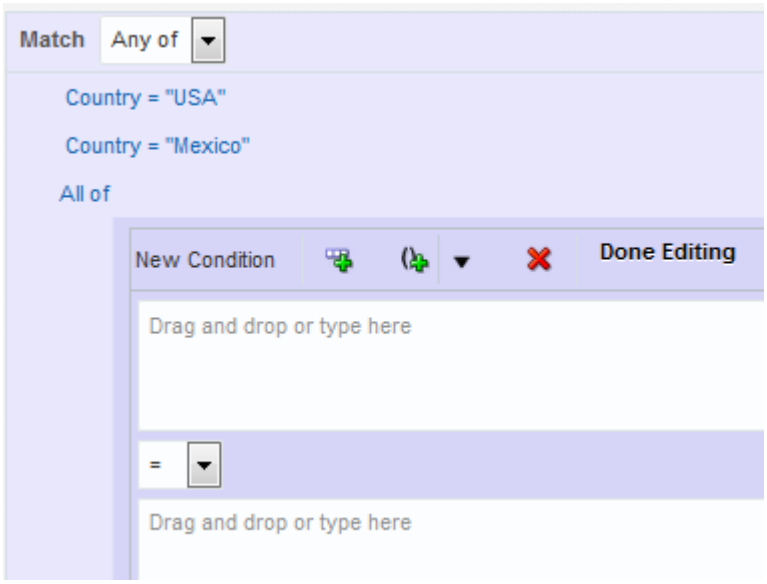
9. Click the **Add Condition** icon (first icon) to add additional condition expressions.

   This creates an additional field for entering additional expression logic. The message **Drag and drop or type here** is displayed in this field.



10. Drag an element from the **Source** area to the first **Drag and drop or type here** field (for this example, the **Country** element is again added).
11. Select an operator (for example, **=**, **>**,**!=**, and so on) and enter a value (for this example, `"Mexico"` is added).
12. From the **Match** list, select an option. This list is hidden until at least two conditions are defined.
    - **Any of**: Select if any of the added expressions must be true. This equates to an OR condition in the entire XPath expression shown in expression mode.
    - **All of**: Select if all expressions must be true. This equates to an AND condition in the entire XPath expression shown in expression mode.

13. Select the **Add Group** icon (second icon) to group a series of conditions. This option enables you to build a number of conditions within a single group. The group is identified by the gray outline and the indentation.



14. Add an element from the **Source** area.

    For this example:

    - The **DisplayName** element is added to the first **Drag and drop or type here** field.
    - The not equal operator (**!=**) is selected.
    - The **Country** element is added to the second **Drag and drop or type here** field.

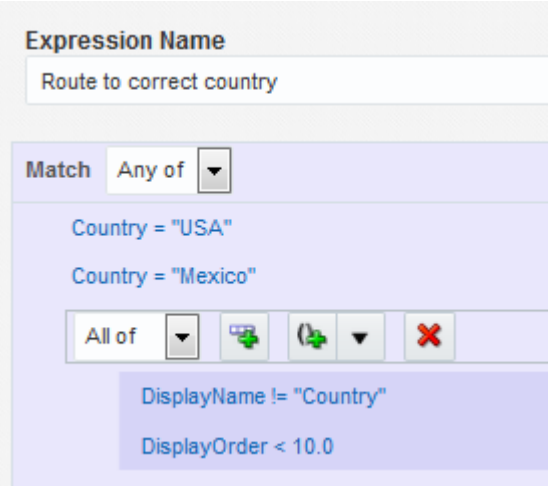
15. Click the **Add Condition** icon (first icon) to add an additional condition expression within the group.

    For this example:

    - The **DisplayOrder** element is added to the first **Drag and drop or type here** field.
    - The less than operator (**<**) is selected.
    - A value of `10` is entered in the second **Drag and drop or type here** field.

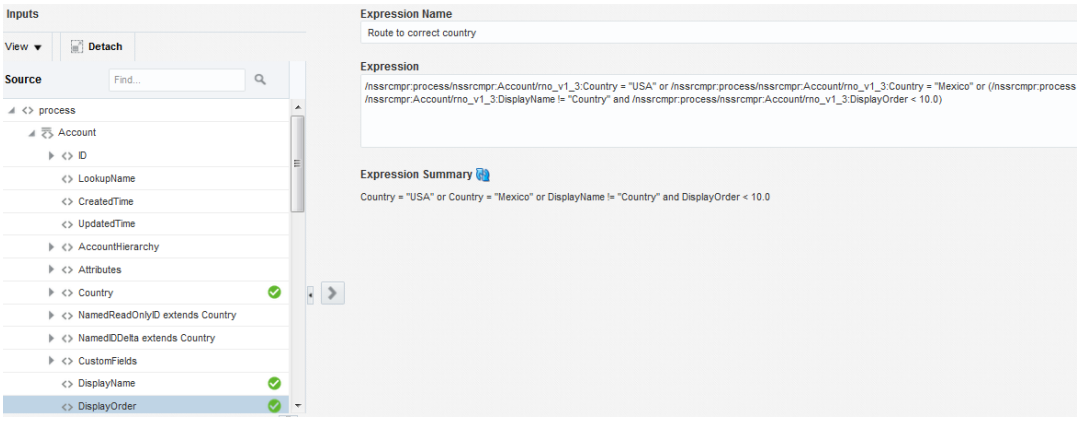
16. Continue building your group condition, as necessary.

    When complete, the expression is displayed. For this example, there are the conditions: if **Country** is **USA** OR **Country** is **Mexico** OR **DisplayName** does not equal country and **DisplayCount** is less than **10**, the integration continues.

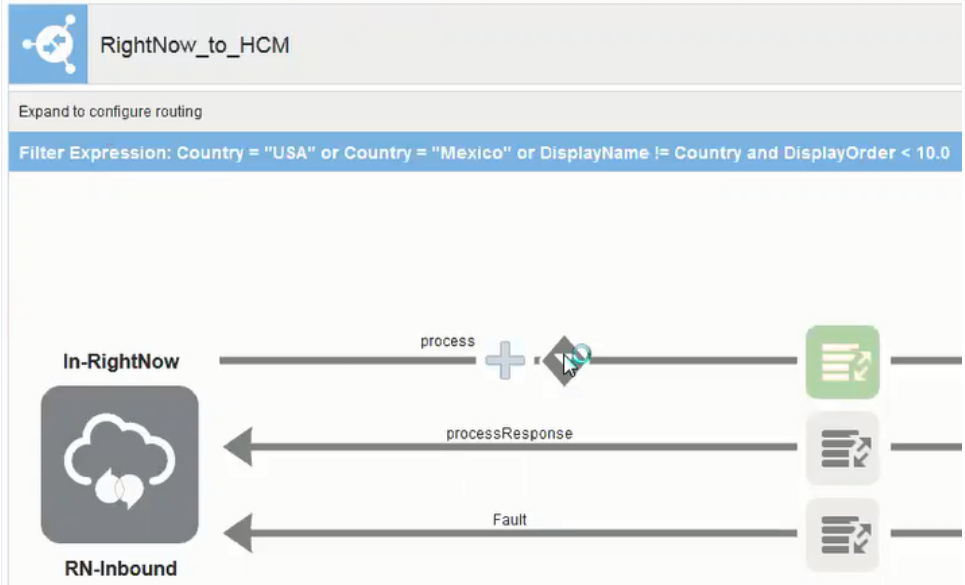


17. Click **Expression Mode**.

    Note the entire XPath expression and the expression summary at the bottom. The selected elements are displayed (no matter their level of depth in the tree) and identified by green checkboxes in the **Source** area.



18. If you want, you can place your cursor in the XPath expression and edit it as necessary (for example, change `USA` to `Canada`), then click the **Expression Summary** icon to refresh the calculation. If you make an error when editing the XPath expression (for example, forget to add a double quote to a value), an error message is displayed.
19. Click **Save** to view the expression in read-only mode. You can also click **Done Editing** at any time during the creation process to view the expression in read-only mode.
20. Click **Close** to return to the integration. The user-friendly expression is displayed in the blue banner above the integration.



Delete Routing Paths

You can delete routing paths that have been created on different target endpoints in an integration.

There are two methods for deleting routing paths:

- Delete the routing path and expression filter.
- Delete the endpoint and routing path, but retain the expression filter.

Deleting the Routing Path and Expression Filter

To delete the routing path and expression filter:

1. In the Integrations page, select the integration in which to delete a routing path.
2. Expand the **Routing Drawer** icon to display the diagram of routing paths.
3. Above the integration, select the routing path to delete.

4. Click the **Filter** icon.
5. Select **Delete Route** from the menu that is displayed.

6. Click **Yes** when prompted to confirm.

   This action deletes the routing path, including the expression filter and the request mapping for the selected path. The diagram above the integration shows that the routing path is deleted.

Deleting the Endpoint and Routing Path

To delete the endpoint and routing path:

1. In the integration, click the target endpoint to delete.
2. Click **Delete** in the menu that is displayed.
3. Click **Yes** when prompted to confirm.

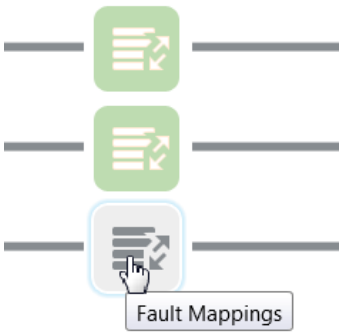
   This action deletes the target endpoint and routing path. The diagram above the integration shows that the routing path is deleted. Within the integration, only the expression remains defined in the integration because it is not using anything from the deleted target endpoint.

Map Faults in Basic Routing Integrations

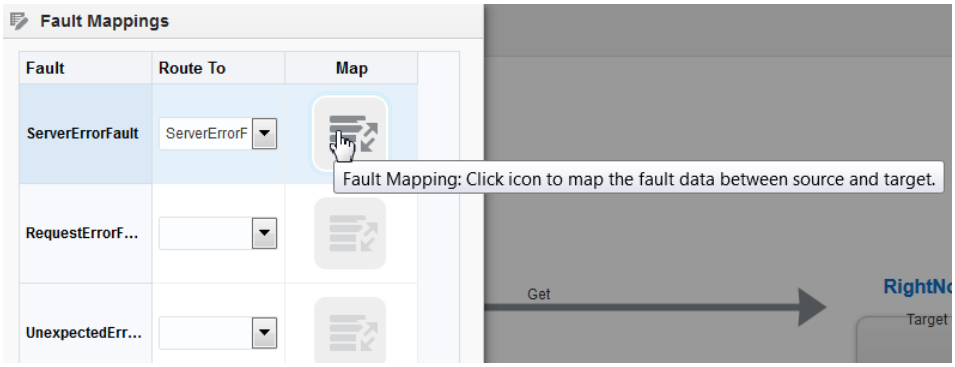
You can map portions of a message into the fault message to compose a description that helps you understand the fault.

To map a fault:

1. Click the **Fault Mappings** icon in an integration.



2. For each fault type, do the following:
   a. Under **Route To**, select the type of fault.
   b. Under **Map**, click the **Mapper** icon of the fault map to perform mapping.



   The mapper appears with the source fault data structure on the left and the target fault data structure on the right. When returning from the mapper, the map icon changes color to indicate it is complete.

3. Click **Close** .
4. Return to the mapping to make any necessary changes to how you mapped your data.

   See Mapping Data of *Using the Oracle Mapper*.

Add Customized Mappings to Prebuilt Basic Routing Integrations

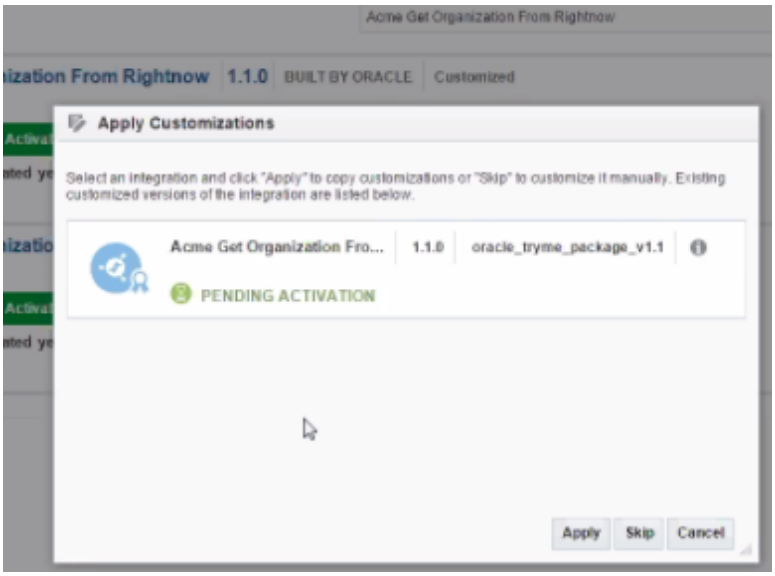
It is a common practice to customize the application endpoints of the prebuilt integrations that you import into Oracle Integration from the Oracle Marketplace (for example, adding custom fields). As a result, you must customize the integration mappings to take advantage of these custom fields. Oracle Integration enables you to customize the mappings in the prebuilt integrations that you import from the Oracle Marketplace. This action creates a customized mapping layer on top of the base mapping file, which is not modified. You can only add customized mappings to prebuilt integrations imported from the Oracle Marketplace, and not to integrations you or another user created.

To add customized mappings to prebuilt integrations:

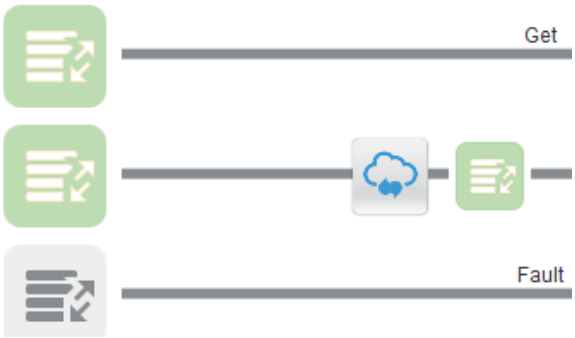
1. In the navigation pane, click **Integrations**.

2. Locate the name of the prebuilt integration to customize. Prebuilt integrations are designated with the words **BUILT BY ORACLE** to the right of the integration name.
3. From the menu at the far right of the integration name, select **Customize**.

   The message **Customizing...** is displayed above the integration.
   If the existence of more than one customized version of the same prebuilt integration is detected, a dialog is displayed that shows a list of versions from which to copy customizations. You can select a version and click **Apply**, or select **Skip** to bypass the copying of customizations and create your own customizations in the mapper, as described in the steps below.



4. Click the icon for the type of mapping you want to customize. You can customize request, response, fault, enrichment source, and enrichment response mappings.

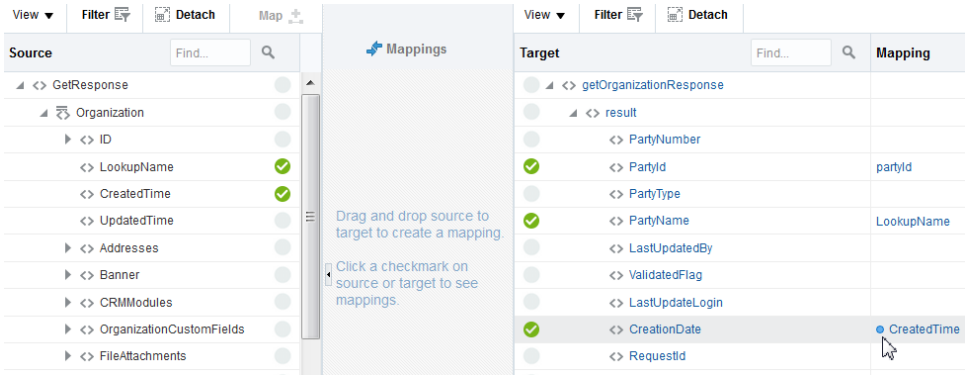


   An icon for customizing the selected mapper is displayed.
5. Click **Customize**.

The mapper is displayed in customization mode.

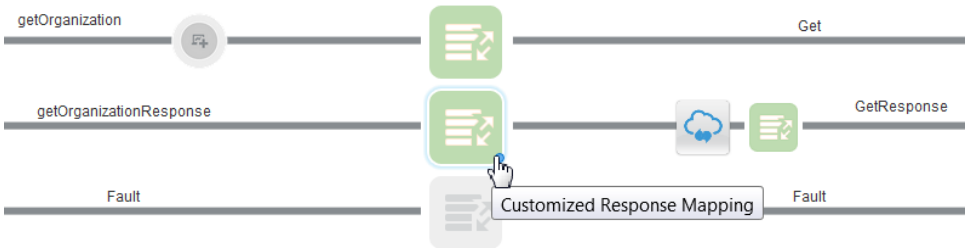
6. Drag and drop source fields to target elements.

   Blue dots are added to the left of the mapped target elements in the **Mapping** column to indicate that these are customized mappings. These mappings are added to a customized layer on top of the base mapping file, which is not modified. This dot differentiates the customized mappings from the regular mappings created as part of the prebuilt integration, which are displayed without a blue dot.



7. Click **Close**, then click **Apply** to save your changes.

   A blue dot with the words **Customized Response Mapping** is displayed in the lower right corner of the icon for the customized mapper (for this example, the response mapper was customized). The other mappers do not have a blue dot because they were not customized (for this example, the request, fault, and request enrichment mappers).



See Mapping Data of *Using the Oracle Mapper with Oracle Integration Generation 2*.

Remove Customized Mappings from Prebuilt Basic Routing Integrations

You can remove the customized mappings that you added to prebuilt integrations that you imported from the Oracle Marketplace. You can remove all customized mappings or specific subsets of mappings (for example, request, response, faults, enrichment source, or enrichment response mappings).

To remove customized mappings from prebuilt basic routing integrations:

1. In the left navigation pane, click **Home** > **Integrations** > **Integrations**.
2. Locate the prebuilt integration in which you want to remove the customized mappings. Prebuilt integrations that have been customized are designated with the words **BUILT BY ORACLE** and **Customized** to the right of the integration name.

3. Click the integration name.

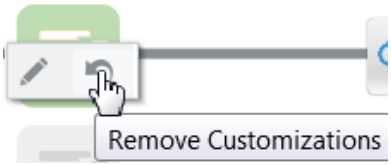
   You can remove all customized mappings added to the integration or specific subsets of mappings (for example, request, response, fault, request enrichment, or response enrichment mappings).

4. To remove all customized mappings from the integration, perform the following step:

   - Click **Remove All Customizations** in the upper right corner.

5. To remove specific subsets of request, response, fault, request enrichment, or response enrichment mappings, perform either of the following steps:

   - Click the mapper icon, then click **Remove Customizations** for the customized mapping to delete (for this example, the customized response mapping is selected).



   or

   a. Click the mapper icon, then click **Customize** to access the specific mapper.

   b. Click **Remove Customizations** in the upper right corner of the mapper page.

6. Click **Yes** when prompted to confirm your selection.

   This action removes the specific customized mappings in the integration. Note that the blue dots that previously identified the customized mappings are removed. The existing mappings that are part of the original prebuilt integration are not removed.

Create Integrations to Publish and Subscribe to Oracle Integration

You can create integrations that enable you to publish messages to Oracle Integration. and integrations that enable you to subscribe to messages from Oracle Integration.

Topics:

Create an Integration to Publish Messages to Oracle Integration

You can create integrations that enable you to publish messages to Oracle Integration. Message publishing is accomplished through use of Oracle Integration Messaging.

To create an integration to publish messages to Oracle Integration:

> **Note:**
>
> Oracle Integration Messaging supports messages of up to 10 MB in size.

1. Follow the steps in Create Integrations to create an integration to publish to Oracle Integration. This creates an integration style with a predefined Oracle Integration Messaging invoke connection that enables you to publish messages to Oracle Integration.
2. In the integration canvas, drag an adapter from the **Connections** panel on the right to the trigger (source) area of the canvas. For this example, an Oracle CX Sales and B2B Service Adapter is selected.

   The Adapter Endpoint Configuration Wizard is displayed.
3. On the Basic Info page, enter an endpoint name and optional identifier for this connection.
4. Click **Next**.
5. On the Request page, select a business object (for this example, **Account** is selected), then click **Next**.
6. On the Response page, select **None** as the response type, then click **Next**.
7. On the Summary page, click **Done**.

   The Oracle ERP Cloud Adapter is configured to publish messages to Oracle Integration through use of Oracle Integration Messaging. Note that there is no request mapper available with this type of integration style.

8. Click **Save**, then click **Close**.

   To subscribe to the message configured in this section, you must now configure Oracle Integration to act as a publisher. This enables Oracle Integration to publish the messages to which other adapters can then subscribe. See Create an Integration to Subscribe to Oracle Integration.

Create an Integration to Subscribe to Oracle Integration

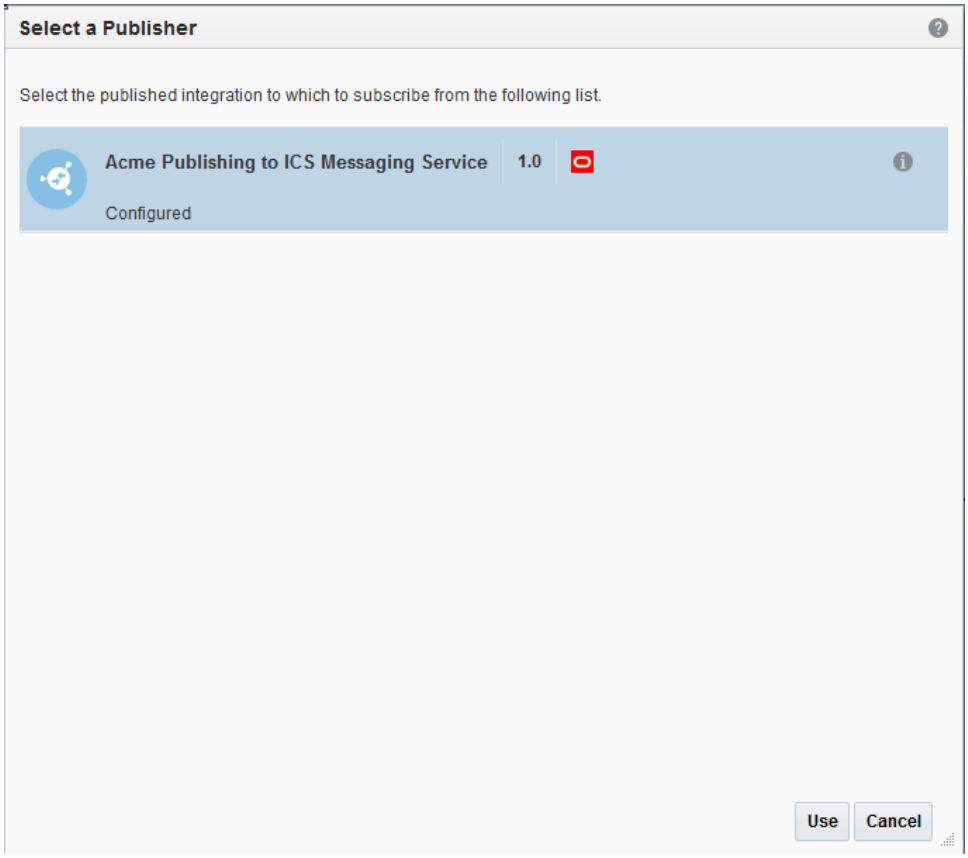
You can create integrations that enable you to subscribe to messages from Oracle Integration. Message subscription is accomplished through use of Oracle Integration Messaging.

To create an integration to subscribe to Oracle Integration:

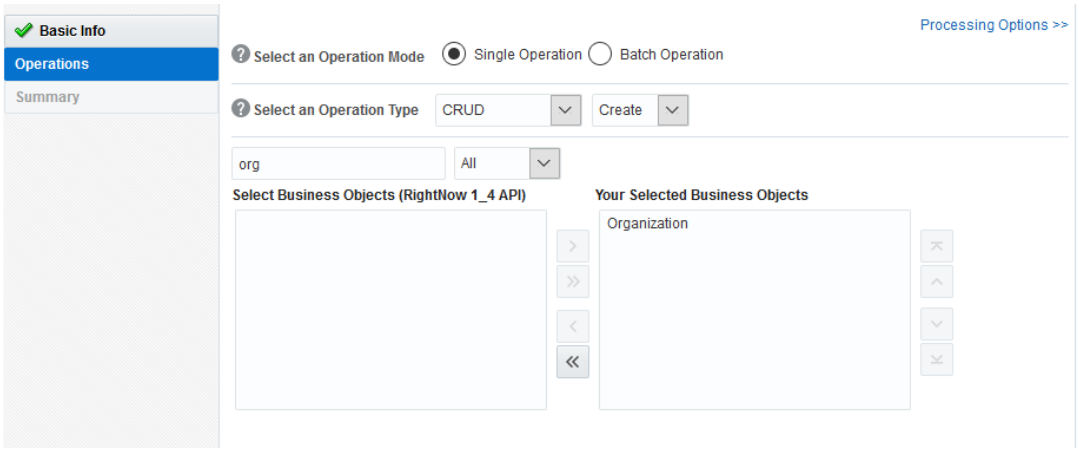
1. Follow the steps in Create Integrations to create an integration to subscribe to Oracle Integration. This creates an integration style with Oracle Integration Messaging that enables you to subscribe to messages from Oracle Integration.

   The Select a Publisher dialog is displayed.

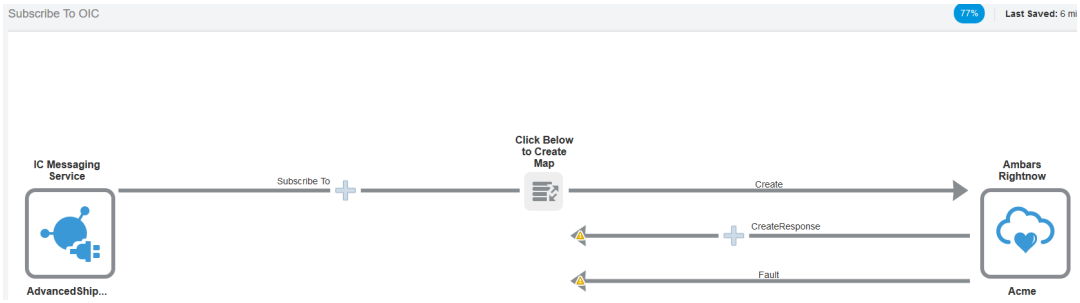
2. Select the integration to which to subscribe, then click **Use**. For an integration to be displayed for selection, you must first configure Oracle Integration as a subscriber, as described in Create an Integration to Publish Messages to Oracle Integration. Only integrations that are 100% completed and unlocked are displayed. Integrations that are locked (meaning that they are being edited) are not displayed.



3. Drag an adapter to the invoke (target) area of the integration designer. For this example, an Oracle Service Cloud (RightNow) Adapter is added.
4. On the Basic Info page, enter a name and optional identifier for this connection.
5. Click **Next**.
6. On the Operations page, select an appropriate operation and business object, then click **Next**. For this example, a CRUD **Create** operation and **Organization** business object are selected.



7. On the Summary page, review your changes, then click **Done**.



The request mapper is available with this type of integration style.

8. Click the **Request Mapping** icon, then click **Create**.
9. Map source fields to the corresponding target fields. See Mapping Data of *Using the Oracle Mapper with Oracle Integration Generation 2*.
10. When complete, click **Close**, then click **Apply** to save your changes.

    The Oracle Service Cloud (RightNow) Adapter is configured to subscribe to messages from Oracle Integration through use of Oracle Integration Messaging.
11. Click **Save**, then click **Close**.
12. Activate the publishing integration described in Create an Integration to Publish Messages to Oracle Integration and the subscribing integration described in this section. See Activate an Integration.

    The completed publishing and subscription integrations enable you to:

    - Create an object in one application that causes the object to be created in other applications.
    - Enable multiple applications to subscribe to Oracle Integration and be registered for updates.
    - Enable additional subscribers to be added or removed without impacting other subscribers or publishers.

Business identifier tracking data is copied when a subscriber is created. If a publishing integration is updated later, you must update the subscribing integration.

For example, assume you create a publishing integration, then create a subscribing integration and select to subscribe to the publishing integration. Select **Tracking** from the  menu, and note that the tracking attributes of the selected publishing integration are displayed. Assume you then edit the publishing integration and change the operation of the trigger adapter (as an example), save, and exit the canvas. If you then edit the subscribing integration and select **Tracking** from the  menu, note that the business identifier tracking attributes of the publishing integration that are displayed are those that existed *before* the updates were made. The tracking fields are not updated as per the updated publisher integration. This is the expected behavior.

7

Map Data and Create Lookups

A visual mapper is provided that enables you to map fields between applications with different data structures by dragging source fields onto target fields. Lookups associate values used by one application for a specific field to the values used by other applications for the same field. This provides the capability to map values across vocabularies or systems.

Topics:

Map Data

Use the mapper to drag fields from the source structure to the target structure to map elements between the two.

Topics:

See Map Data of *Using the Oracle Mapper with Oracle Integration Generation 2*.

Access the Mapper

To create mappings in an integration, you need to first access the mapper. The method for accessing the mapper is based on the integration style you are using.

See Access the Mapper in *Using the Oracle Mapper with Oracle Integration Generation 2*.

Video

Encode and Decode File Attachment Content

The virtual file system (VFS) enables you to store files and internally use references to these files in the message payload. You can also map the VFS file's content to a string element.

For example, you can store files and use references in the VFS as follows:

- The REST Adapter supports the multipart attachment and application/octet-stream features. The attachment is stored in a staging area and an `attachmentReference` (string key) is generated. The `attachmentReference` key is sent as part of the message payload and later fetches the attachment instance from the staging area.
- The FTP Adapter uses `fileReference` for reading/writing a file without a schema. `fileReference` is also a reference to a file stored in the VFS.

In addition, mapping the VFS file's content to a string element enables you to:

- Map the content of a staged file attachment to a string element by converting the content to a base64 string.
- Store the base64 string as an attachment and generate a VFS reference.

Two XPath functions are provided to perform these tasks. These functions work with any adapter.

- `encodeReferenceToBase64(String reference)`: Accepts the VFS's file reference as input and returns the base64–encoded content of the file as the return value. This function has a file size limit of 10 MB. If a file is larger than 10 MB, an exception message of `Maximum file size supported is 10 MB` is displayed.
- `decodeBase64ToReference(String base64String)`: Accepts the base64–encoded content as input, decodes it, stores the base64–decoded value in a file in the VFS, and returns the reference to this file. There is no size limit because the content is already in memory.

  The `77u/` text at the start of an encoded base64 string is related to the byte order mark and indicates the encoding scheme used (UTF-8). This is not needed for decoding of base64 string to a binary.

  This is needed for representing binary data in a string. This text must be extracted and removed from the original base64 string to ensure it doesn't cause decoding issues. If this is not removed from the encoded string, the decoded data, if converted to a string, has additional space.

  You can extract and remove this from the base64-encoded string with the help of standard mapper functions for strings. This information is not needed to decode the base64-encoded string to binary data. This is needed for reading the binary data using UTF-8 encoding.

The location is the relative path (reference) of the file stored in Oracle Integration. The relative path is one of the following elements:

- `fileReference`
- `attachmentReference`
- `streamReference`

The two XPath functions are available for use in Oracle Integration:

- Expression Builder, when configuring the following actions in an orchestrated integration:
  - Notification
  - Logging
  - Switch
  - Assign
- Mapper (visible after selecting  > **Functions** > **Advanced**)

The attachments are not restricted to document file types. For example, an image can be base64–encoded and later decoded back to the original file.

When an attachment is stored in the VFS, a key is generated to retrieve the attachment at a later time. The key is shown in the mapper as `attachmentReference/fileReference/streamReference`. This key is propagated within Oracle Integration as part of the payload. The attachment is claimed only when needed. The names `attachmentReference`, `fileReference`, and `streamReference` are based on the adapter type. For example, in the REST Adapter, `streamReference` is used. The data type of the reference is a string.

With a multipart feature, the HTTP request payload has multiple parts separated by boundaries. Each of the individual parts are considered an attachment. For raw bytes, `streamReference` is used. FTP uses `fileReference`.

Sometimes the endpoints accept only base64–encoded values. In these cases, the reference is passed as input to `encodeReferenceToBase64` to get the base64–encoded content of the file. Again, the base64–encoded value can be passed as input to `decodeBase64ToReference` to get the reference (location) to a file that contains the decoded content.

Import Map Files

Review the following topics to learn how to import map files into Oracle JDeveloper and Oracle Integration.

> **Note:**
>
> As an alternative, you can directly edit XSLT code in the mapper. This eliminates the need to export your XSLT code from Oracle Integration, edit the code manually in a text editor or in a separate graphical tool such as Eclipse or Oracle JDeveloper, and then re-import the code into the mapper in Oracle Integration. See Edit the XSLT Code in the Mapper in *Using the Oracle Mapper with Oracle Integration Generation 2*.

Topics:

- Import a Map File into Oracle JDeveloper
- Import a Map File into Oracle Integration

You can export an integration that includes a map file that you want to edit in Oracle JDeveloper. See Export an Integration.

Import a Map File into Oracle JDeveloper

You can import an Oracle Integration archive file into an Oracle Service Bus project in Oracle JDeveloper. The archive file can include a map file that is largely complete in content or a map file that is empty of content. This action enables you to perform advanced XSLT tasks (create variables, use templates, and so on) in Oracle JDeveloper that you cannot perform in the Oracle Integration mapper. After you complete these advanced tasks in Oracle JDeveloper, you can save and re-import the map file into Oracle Integration.

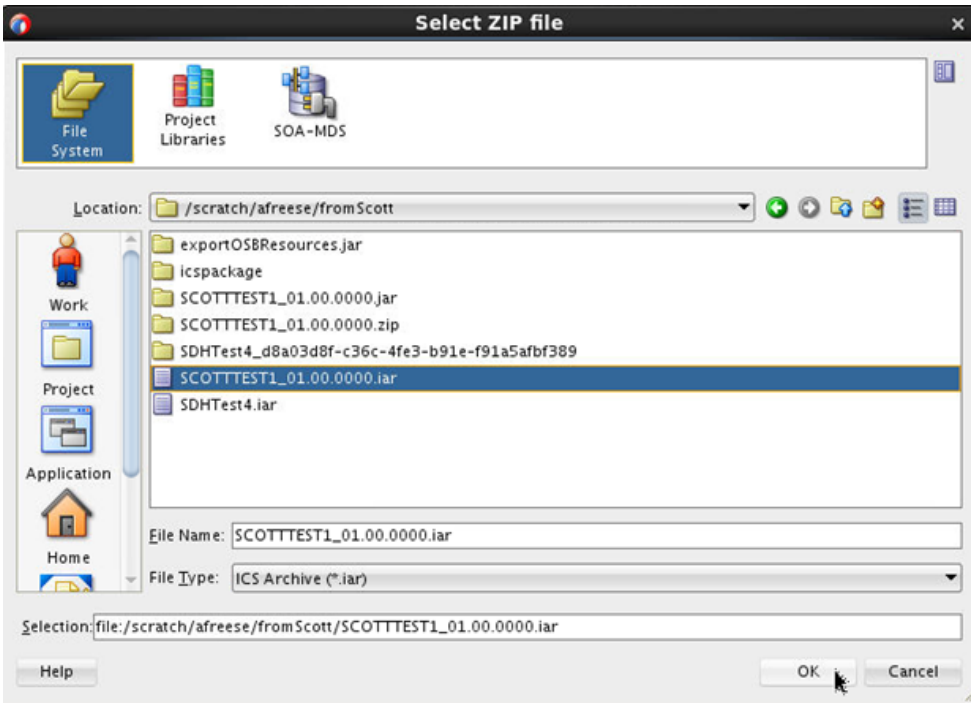
1. See Export an Integration for instructions on exporting an integration that includes the map file you want to edit in Oracle JDeveloper.
2. Create an Oracle Service Bus application with a project in Oracle JDeveloper.
3. In the application navigator, right-click the Oracle Service Bus project and select **Import**.

   The Import dialog is displayed.
4. Select **Service Bus Resources**, and click **OK**.

   The Import Service Bus Resources wizard is displayed.
5. Select **Zipped/Archived Resources**, and click **Next**.
6. Click the **Browse Zip Source** icon to the right of the **Zip Source** field.

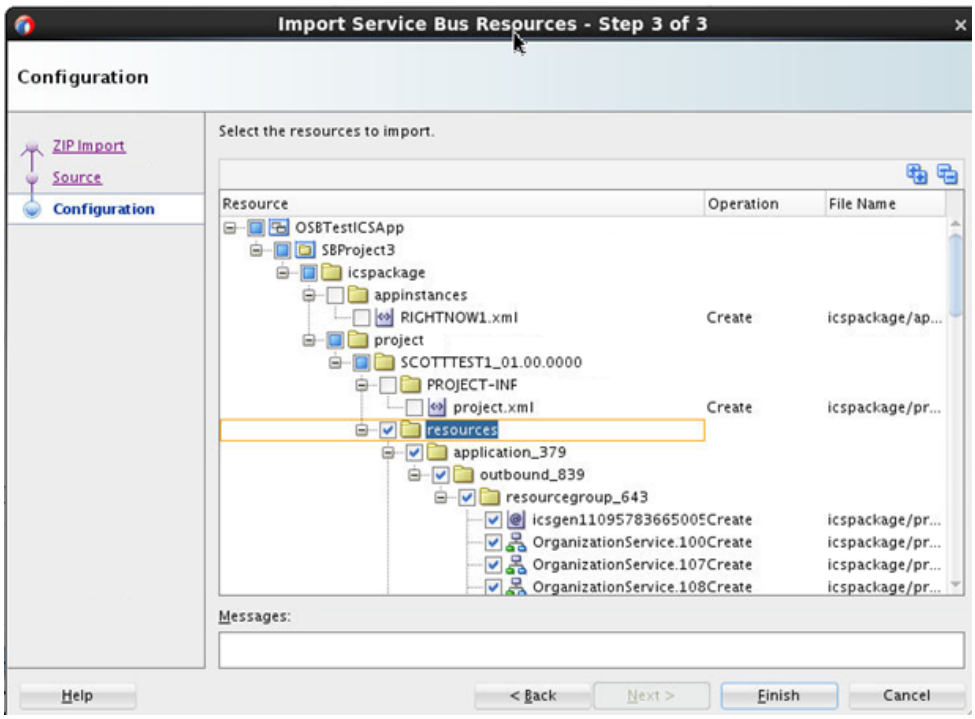
   The Select ZIP File dialog is displayed.

7. If using Oracle JDeveloper 12.2.1.*x*, perform the following steps:
   a. From the **File Type** menu, select **ICS Archive (*.iar)**.
   b. Browse for and select the Oracle Integration IAR archive file that you previously exported.



8. If using Oracle JDeveloper 12.1.3, perform the following steps:
   a. Ensure that you first rename the .iar file extension to `.zip`.
   b. Browse for and select the ZIP file to import.
9. Click **OK**, then click **Next** on the wizard page.

   The contents of the JAR file are displayed and can be selected for import.
10. Select the **resources** folder in which to import the archive file. Note that the entire **Resource** tree is selected by default, including everything above the hierarchy node that you want to select. Ensure that you deselect the parts above the relevant hierarchy node, then click **Finish**.



The resources are imported into the Oracle Service Bus project. You can now open the map file for editing with the XSLT Map Editor in Oracle JDeveloper.

Import a Map File into Oracle Integration

There may be scenarios in which you need to perform an advanced XSLT task (create variables, use templates, and so on) that you cannot perform in the Oracle Integration mapper. For these cases, you can export the integration, import the integration into Oracle JDeveloper, perform these advanced tasks in the map file in the XSLT Map Editor in Oracle JDeveloper, and then save and re-import the map file into Oracle Integration. The map file must be from an Oracle Service Project in Oracle JDeveloper.

1. In the left navigation pane, click **Home** > **Integrations** > **Integrations**.
2. Click the specific integration in which to import the map file.
3. Click the mapper icon to display a menu.
4. Click **More Actions** > **Import**.
5. Click **Browse** to select the map (`.xsl`) file. Note that while you exported the entire integration, you do not import the entire integration back into Oracle Integration. You only import the map file of the exported integration back into Oracle Integration.
6. Click **Import**.

Manage Lookups

A lookup associates values used by one application for a specific field to the values used by other applications for the same field. This provides the capability to map values across

vocabularies or systems. For example, you can map country codes, city codes, currency codes, and so on.

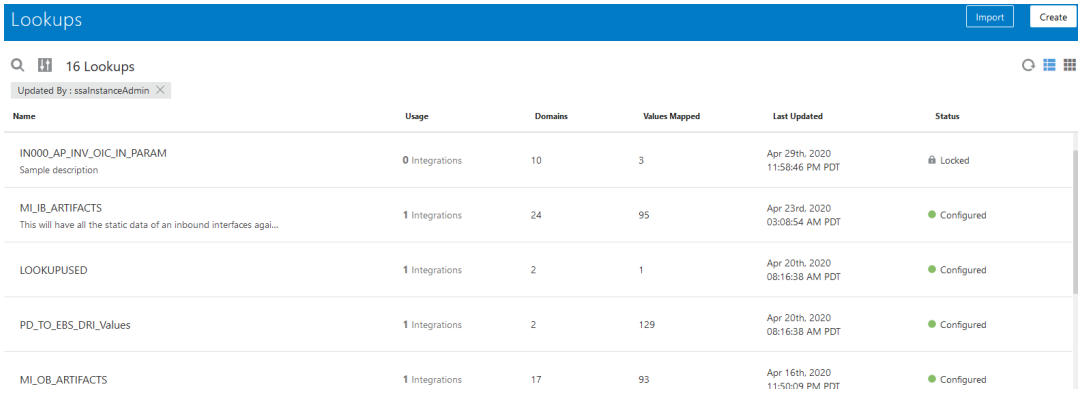
> **Note:**
>
> Specifying a lookup value that includes a special character of # is *not* supported.

Topics:

Create a Lookup

Create a lookup to map values between applications.

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



2. Click **Create**.

   The Create New Lookup dialog is displayed.

3. Enter a name and optional description for the lookup.
4. Click **Create**.

The Lookup page is displayed.

Add Adapters or Domain Names to a Lookup

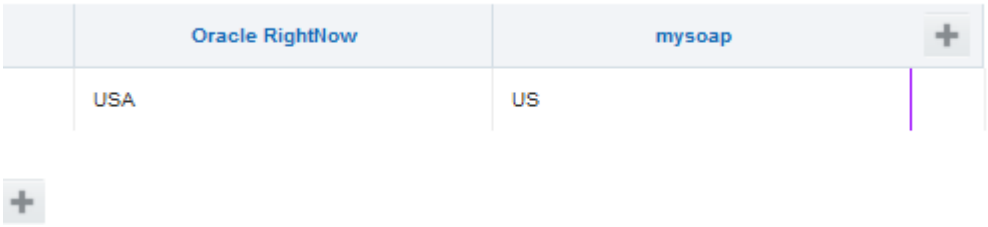
Add adapters or domain names to a lookup to map values between connections.

1. Click **Domain Name 1**.

A menu with options is displayed.

2. Select the type to create:
   - **Replace with Adapter**: This selection invokes the Select Adapter dialog for choosing the adapter to use. You can change your selection:
     - If you want to change your adapter selection, click the selected adapter type, then choose **Select Different Adapter**.
     - If you want to change from an adapter to a domain name, click the selected adapter type and choose **Replace with Domain Name**.
   - **Enter Domain Name**: This selection invokes a dialog for entering the domain name to use. There is a 50 character limit. You can change your selection:
     - If you want to change from a domain name to an adapter, click the domain name, then choose **Replace with Adapter**.
     - If you want to change the domain name, click the name, then choose **Edit Domain Name**.

     The adapter or domain name is displayed.
3. Enter a value in the field below the adapter or domain name. To add more rows for additional values, click the **+** sign.
4. Click **Domain Name 2**.
5. Repeat Step 2 to add an adapter or domain name.
6. Enter a value in the field below the adapter or domain name. To add more rows for additional values, click the **+** sign.



7. If you want to add another column, click **+**.
8. Repeat to add more adapters and domain names to the lookup.
9. Click **Save** when complete.

Clone a Lookup

You can clone a copy of an existing lookup. It is a quick way to create a new lookup with similar information. Lookups with adapters and domain names can be cloned.

1. In the navigation pane, click **Home** > **Integrations** > **Lookups**.
2. Go to the row of the lookup to clone.
3. On the Lookups page, select **Clone** from the

menu.

The Clone Lookup dialog is displayed.

4. Enter the lookup information.
5. Click **Clone**.
6. Click **Edit** to further configure your cloned lookup.

Delete a Lookup

You can delete a connection from the Lookup menu.

1. In the navigation pane, click **Home** > **Integrations** > **Lookups**.
2. Go to the row of the lookup to delete.
3. Select **Delete** from the 
menu.

   The Delete Lookup dialog is displayed.
4. Click **Yes** to confirm deletion.

Update Lookups in Active Integrations

Lookups are not deployed as part of integration activation. Therefore, changes that you make to lookups already used in active integrations typically take effect immediately. There is *no* need to re-activate integrations using a changed lookup for the new lookup value to take effect.

There can be instances in which updates to the runtime environment are slow because it is not in real-time and queuing can occur at runtime. For those cases, updates can occur in a couple of minutes.

Export a Lookup

Once you create a lookup, you can export that lookup for use in other Oracle Integration environments. You can export a lookup from either the Lookup Designer or from the Lookups list.

To export a lookup:

1. In the navigation pane, click **Home** > **Integrations** > **Lookups**.
2. Go to the row of the lookup to export.
3. Select **Export to CVS** from the 
menu.
4. In the dialog that appears, select **Save File**, and then click **OK**.
5. Save the file to the location you want.

The file is saved as a CSV file with the same name as the lookup.

Import a Lookup

You can import lookups that were previously exported from Oracle Integration.

The file to import must have the following for the first row, where *table_name* is the name of the table as you want it to appear in Oracle Integration. This name cannot contain spaces.

```
DVM,table_name
```

The second row contains the names of the adapters that are being mapped. Use the following case-sensitive IDs for each adapter:

Adapter Name	Identifier to Use in the Import File
Oracle Eloqua Cloud	eloqua
Oracle Engagement Cloud	osc
Oracle Messaging Cloud Service	oms
Oracle Service Cloud	rightnow
Oracle HCM Cloud	hcm
Oracle ERP Cloud	erp
Salesforce Cloud	salesforce

To import a lookup:

1. Locate the CSV file containing the lookup table you want to import.
2. In the navigation pane, click **Home** > **Integrations** > **Lookups**.
3. In the banner, click **Import**.
4. In the Import Lookup dialog, click **Browse** to navigate to and select the CSV file to import.
5. Click **Import**.

   If a lookup already exists with the same identifier and version, you must confirm whether to overwrite the existing lookup.

The imported lookup appears in the Lookups list on the left. You can customize or activate it, if it is ready.

Create the lookupValue Function

You can create the parameter values for the `lookupValue` function with the Build Lookup Function wizard. This wizard enables you to define the lookup table, source column, target column, and default value to use in the function. For these parameter values to be selectable in the wizard, you must have already created a lookup on the Lookups page.

Topics:

Access the Build Lookup Function Wizard

The Build Lookup Function wizard for creating the `lookupValue` function parameter values is accessible from the mapper or the Expression Builder in Oracle Integration.

To access the Build Lookup Function wizard:

> **Note:**
>
> You must already have created lookups to use this wizard. See Manage Lookups.

- From the mapper:
  See Access the Build Lookup Function Wizard.
- From the Expression Builder in an action, such as the switch action:
  1. Go to the Integrations page.
  2. Open an integration.
  3. Invoke the Expression Builder from within the integration. For example:
     - In an app-driven orchestrated integration, edit a switch activity.
     - In a basic routing integration, click the **Filter** link.

     The Expression Builder is displayed.
  4. Expand **Components**.
  5. Expand **Functions** > **Oracle Integration**.
  6. Expand the `lookupValue` function to view the available parameters to define in the Build Lookup Function wizard and Expression Builder.
     - `dvmLocation`
     - `srcColumn`
     - `srcValue`
     - `targetColumn`
     - `defaultValue`
  7. Drag the `lookupValue` function into the **New Condition** field.

     The Build Lookup Function wizard is displayed. To create the `lookupValue` function parameter values, see section Select the Lookup Table.

Select the Lookup Table

Select the lookup table to use in the `lookupValue` function.

> **Note:**
>
> You must already have created a lookup. Otherwise, no lookups are displayed for selection.

Element	Description
Lookup Table	Select the lookup table to use in the function. You can view the lookup description by clicking the information icon in the table. This can guide you in selecting the required lookup table. The number of columns defined in the lookup is also displayed.

Select the Source and Target Columns

Select the source and target columns to use in the `lookupValue` function.

The `lookupValue` function requires one source column and one target column. When you select a source and target column, the values available with the columns are displayed.

Element	Description
Select Source Column	Click the source column header to select from a list of available columns for this lookup table. The data included with the selected column is displayed. Both adapter and domain name columns are displayed.
Select Target Column	Click the target column header to select from a list of available columns for this lookup table. The data included with the selected column is displayed. Both adapter and domain name columns are displayed.

Specify the Default Value

Select the default value to use in the `lookupValue` function.

Enter the default value to use if no match is found. If there is no match that satisfies all the search values, the lookup fails and the default value is returned.

Element	Description
Default Value	Enter a default value to use if no match is found (for example, an actual default value to use or an error message such as `No Value Found`).

Review Your Lookup Table Selections

You can review the lookup table values to use in the `lookupValue` function on the Summary page.

You can review the lookup table values from the Summary page. The Summary page is the final wizard page after you have completed your configuration.

Element	Description
Parameter and Value Table	Displays a summary of the parameters and values you defined on previous pages of the wizard. To return to a previous page to update any values, click the appropriate tab in the left panel or click **Back**.

Resulting Expression	Displays the expression you defined on the previous pages of the wizard. The `lookupValue` function takes the following format:
	`lookupValue(dvmLocation, srcColumn, srcValue, targetColumn, defaultValue)`
	Where:
	• `dvmLocation`: The lookup table selected on the Select Lookup Table page.
	• `srcColumn`: The source column selected on the Select Columns page.
	• `srcValue`: The source value you enter in the **New Condition** field of the Expression Builder after completing this wizard. Click **Done** to complete this wizard, then define the `srcValue` parameter value.
	• `targetColumn`: The target column selected on the Select Columns page.
	• `defaultValue`: The default value entered on the Default Value page.
	For example, a defined `lookupValue` function after you have completed the wizard and defined the `srcValue` parameter value in the Expression Builder can look as follows:
	`dvm:lookupValue('tenant/resources/dvms/Country','rightnow','US','mysoap','No data found')`

When you click **Done**, the **function** icon is created in the mapper and the function XPath expression is displayed in the Expression Builder.

8

Manage the Agent Group and the On-Premises Connectivity Agent

You must create an agent group and install the on-premises connectivity agent before you can create an integration in which messages are exchanged between Oracle Integration and your locally-installed, application resources residing in an on-premises network or a private cloud.

Topics:

Conceptual details and patterns for using the connectivity agent are provided. See About Creating Hybrid Integrations Using Oracle Integration .

If you experience problems with the on-premises connectivity agent, troubleshooting information is provided. See Troubleshoot the Connectivity Agent.

Download and Run the Connectivity Agent Installer

You must create an agent group and download and run the connectivity agent installer to install the agent in your local environment.

Create an Agent Group

You must create an agent group in Oracle Integration before you can run the connectivity agent installer. When you install the connectivity agent in your environment, you associate the connectivity agent with the agent group identifier. Up to two connectivity agents can be associated with an agent group. For a single Oracle Integration instance, you can create up to five agent groups. Creating the agent group also creates the necessary artifacts required for message exchange.

To create an agent group:

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

2. Click **Create Agent Group**.

   The Create New Agent Group is displayed.

3. Enter the following information, then click **Create**.

Field	Description
Name	Provide a meaningful name so that others can understand the agent name. The name must be unique among all agent names in the system. The name can consist of the following: • Letters (A-Z, a-z) • Numbers (0-9) • Spaces ( ) • Special characters (_ -) The name must not begin or end with a space and cannot be longer than 50 characters.
Identifier	Accept the default identifier value or change it, if necessary. The identifier is initially the same as the agent group name you provided, but in upper case. When you install the agent, you must specify the identifier value. **Note**: After creating the agent group, you cannot edit the agent group identifier. Instead, you must delete and recreate another agent group to associate with a different agent group identifier.
Type	**Connectivity Agents Group** is displayed and cannot be edited. The connectivity agent supports integrating with on-premises systems. The agent group references only connectivity agents.
Description	Provide a meaningful description so that others can understand the responsibilities of the agent group.

System Requirements

Satisfy the following system requirements before installing the on-premises connectivity agent.

- Understand the JDK Version, Operating System, and Heap Size Requirements
- Ensure the Connectivity Agent Version is Compatible with JDK 17
- Upgrade the Existing JDK 11 Agent to the JDK 17 Agent
- Revert Back to the JDK 11 Agent in Case of Issues

Understand the JDK Version, Operating System, and Heap Size Requirements

- Install and use JDK version 17.

> **Note:**
>
> To comply with Oracle security standards, JDK 8 and JDK 11 are being deprecated for use with the on-premises connectivity agent. You must upgrade to JDK 17 as soon as possible and convert the JKS keystore to the PKCS12 keystore at the same time. See Complete Upgrade Prerequisites in *Provisioning and Administering Oracle Integration 3*.
> Note the following key deadlines:
>
> – Support for the connectivity agent on JDK 8 and JDK 11 ends on December 31, 2023. If you continue to use either version, you no longer receive quarterly updates for the connectivity agent and support for any submitted service requests that involve the connectivity agent.
>
> – Some services that depend on the connectivity agent may not function properly after January 15, 2024. This means that connectivity agent traffic on JDK 8 and JDK 11 may be blocked by Oracle.
>
> The new announcements banner on the Home page also provides information about the JDK 8 and 11 deprecation.
>
> > **Note:**
> >
> > The announcement banner does *not* disappear after upgrading to JDK 17 and the PKCS12 keystore. To dismiss the banner:
> >
> > 1. In the navigation pane, click **Settings**, then **Upgrade**.
> > 2. Run the upgrade eligibility program regardless of whether you are preparing to upgrade.
> >
> > The banner goes away shortly.

Note the following details:

– There are JDK 11 and JDK 17 restrictions when using the Oracle WebLogic JMS Adapter. See Connectivity Agent Restrictions.

– If you are using JDK 8 with the connectivity agent, ensure that JDK 8 includes the latest updates. After applying these updates, restart the agent. Certificates in older versions of JDK 8 can expire, which causes connectivity issues for the agent.
Move to JDK 17. JDK 17 is the minimum version required for upgrading the agent from Oracle Integration Generation 2 to Oracle Integration 3. Customers should plan to uptake the latest JDK patches periodically after they move to JDK 17.

– You do not need to purchase a JDK license for the sole purpose of installing and/or running the agent. See JDK license and support for OIC Connectivity Agent (Doc ID 2611142.1).

– The JDK installation can be the same one as used with other products installed on the same host. However, separate JDK licensing may be required for any use of Java beyond the use of the agent. Ensure that the JDK installation is *not* modified for use with these other products.

- Install the connectivity agent on one of the following certified operating systems:

  – Oracle Linux 6.*x*

  - Oracle Linux 7.*x*
  - Oracle Linux 8.*x*
  - RedHat Enterprise Linux 6.6
  - RedHat Enterprise Linux 7.*x*
  - RedHat Enterprise Linux 8.*x*
  - Suse Linux Enterprise Edition 12 SP2
  - Windows Standard Edition 2016
  - Windows 2019

> **Note:**
>
> - The connectivity agent is not certified on Kubernetes (K8s).
> - IBM or Open JDK are *not* supported.

- Provide a minimum of 8 GB memory with 4 GB of heap size dedicated for the agent JVM. If you want to include any other processes on that host besides the on-premises agent, it is strongly recommended that you increase physical memory to a value greater than 8 GB.

Ensure the Connectivity Agent Version is Compatible with JDK 17

Run the following command from the `agenthome` directory to determine if the connectivity agent is compatible with JDK 17. The following minimum version on Oracle Integration Generation 2 is required for JDK 17 compatibility.

```
$ cat version
230505.0528.4033
```

If you receive the following error when starting the connectivity agent after upgrading to JDK 17, you are not running a compatible version of the connectivity agent:

```
Agent is only supported on Java version 8 or 11. Please set JAVA_HOME and
PATH to the location
for Java (JDK 8 or 11). This error occurs because you are using an older
version of the connectivity agent.
```

You may be running an older version of the connectivity agent for the following reasons:

- Previous attempts to upgrade the connectivity agent failed
- You started the agent with the following option:

  ```
  -DenableAutoUpgrade=false
  ```

To resolve this issue:

1. Start the connectivity agent once with your previous JDK version (JDK 8 or JDK 11).
2. Wait for the connectivity agent upgrade to finish. You can monitor upgrade in the `agenthome/logs/agent-upgrade.log` file.
3. Stop the connectivity agent.

4. Restart the connectivity agent with JDK 17.

Upgrade the Existing JDK 8 or JDK 11 Agent to the JDK 17 Agent

1. Install JDK 17.
2. Stop the existing connectivity agent.
3. Set `JAVA_HOME` and `PATH` to `JDK17`.
4. Restart the connectivity agent.

Revert Back to the JDK 8 or JDK 11 Agent in Case of Issues

1. Stop the connectivity agent.
2. Set `JAVA_HOME` and `PATH` to `JDK8` or `JDK11`.
3. Restart the connectivity agent.

Connectivity Agent Restrictions

Note the following connectivity agent restrictions.

- When the Oracle WebLogic JMS Adapter is used with a version of Oracle WebLogic Server prior to 12.2.1.4, JDK 11 and JDK 17 are not supported. To know your version of Oracle WebLogic Server, perform the following steps:
  1. Enter the Oracle WebLogic Server URL. For example:

     ```
     http://abcweblogic.oracle.com:7001/console
     ```

  2. At the bottom left, find the version. For example:

     ```
     WebLogic Server Version: 12.2.1.4.0
     ```

- When installed on a Windows host, the connectivity agent does not work with a proxy using NT LAN Manager (NTLM) authentication.

Agent Download and Installation

The agent must be downloaded and installed on your local host. If needed, you can also install a security certificate on your local host. If necessary, you can also run the agent installer as a background process.

- Download and Install the Agent
- Install a Certificate on the Agent Host
- Run the Connectivity Agent Installer as a Background Process on Linux Systems

Download and Install the Agent

Install the connectivity agent installer to install the agent in your local environment. During installation, you associate the connectivity agent with the agent group identifier you generated when creating an agent group in Oracle Integration.

> **Note:**
>
> - You must have the Java execute permission to install and restart the agent.
> - You must have the *Service*Administrator role to download the agent.
> See What Users Can Do in the Integrations Design Section by Role of *Provisioning and Administering Oracle Integration Generation 2*.

1. Create a directory for connectivity agent installation on your on-premises host.

   > **Note:**
   >
   > - Do not install the agent in a directory path that includes `/tmp`.
   > The agent must never be installed in `/tmp`. As per the Filesystem Hierarchy Standard version 3.0, `/tmp` is meant for temporary files. Even though the install and agent work, it is not a recommended location for agent installation because the directory in `/tmp` may be deleted after the reboot of the system or agent virtual machine.
   > - Agent installation is not supported with use of an SSL proxy.

2. In the left navigation pane, click **Home** > **Integrations**, then click **Agents**.
3. Click **Download** > **Connectivity Agent**.
4. Download the connectivity agent installer to the directory on your on-premises host.
5. Unzip `oic_connectivity_agent.zip`.
6. If you need to add any third party JARs (for example, for the Siebel Adapter or MySQL Adapter), copy them under the `agenthome/thirdparty/lib` directory.

   > **Note:**
   >
   > If you perform this step *after* installing the connectivity agent, you must restart the agent.
   > See Restart the Agent.

7. Modify `InstallerProfile.cfg` to include the following information:

```
# Required Parameters
# oic_URL format should be https://hostname:sslPort
oic_URL=https://oic_host:ssl_port
agent_GROUP_IDENTIFIER=
```

```
#Optional Parameters
oic_USER=
oic_PASSWORD=

#Proxy Parameters
proxy_HOST=
proxy_PORT=
proxy_USER=
proxy_PASSWORD=
proxy_NON_PROXY_HOSTS=
```

Where:

Parameter	Description
`oic_URL`	This parameter is required. This is the HTTPS URL for the Oracle Integration host. The port is `443`.
`agent_GROUP_IDENTIFIER`	This parameter is required. This is the identifier for the connectivity agent group created in Oracle Integration. The identifier name is case sensitive. See Create an Agent Group.
`oic_USER`	This optional parameter provides the Oracle Integration user name. When the agent runs for the first time, this field, if provided, is encrypted in the properties file. If this field is not provided, you are prompted to enter the user name at agent startup and it is not persisted. See What Users Can Do in the Integrations Design Section by Role of *Provisioning and Administering Oracle Integration Generation 2*. The connectivity agent supports basic authentication. You cannot use Oracle Cloud Infrastructure API key or OAuth parameters.
`oic_PASSWORD`	This optional parameter provides the Oracle Integration password. When the agent runs for the first time, this field, if provided, is encrypted in the properties file. If this field is not provided, you are prompted to enter the password at agent startup and it is not persisted.

Parameter	Description
`Proxy Parameters`	These parameters are only required if the connectivity agent is used with a proxy in the on-premises environment. • If you have multiple hosts that must be configured in a nonproxy host environment, you must separate each IP address or host with a pipe symbol (`|`) in the `proxy_NON_PROXY_HOSTS` parameter. For example: `proxy_NON_PROXY_HOSTS=localhost|127.0.0.1|*.myorg.com` • If you use a proxy user for Windows, the user name must include the MS domain name in front of the user name, along with double backslashes before the user name (for example, *MS_domain**username*). If you do not specify the double backslashes, you receive a `407 Proxy Authentication Required` error.

8. Set the `JAVA_HOME` property to the location of the JDK installation.
9. Set the `PATH` property. For example, if `csh` is your shell:

   ```
   setenv PATH = $JAVA_HOME/bin:$PATH
   ```

10. Run the connectivity agent installer from the command prompt. If you want to copy and paste this command, ensure that it does not have any special characters.

    ```
    java -jar connectivityagent.jar
    ```

11. Provide the Oracle Integration credentials when prompted.

    ```
    Proceeding to install a new agent ...
    Enter your OIC username :
    Enter password for username :
    ```

12. Wait for a successful installation message to appear.

    ```
    Done with Agent installation & configuration... Starting agent for message
    processing.
    Agent started successfully... listening for new messages...
    ```

    If errors occur, review the agent diagnostic logs.

    See Monitor Agents.

13. Depending on your agent environment, you may also need to install a certificate.

    See Install a Certificate on the Agent Host.

Install a Certificate on the Agent Host

If you need to add a certificate on the agent host, use the `keytool` to import the certificate in `keystore.jks`. Installing the certificate enables you to access hosts with self-signed certificates. It is not normally needed.

> **Note:**
>
> If you install a certificate *after* installing the connectivity agent, you must restart the agent.
> See Restart the Agent.

Scenarios under which you need to import the certificate in the agent keystore are as follows:

- The connectivity agent is used with an SSL proxy.
- The connectivity agent is used to invoke secure (SSL) on-premises endpoints.

1. Go to the `agenthome/agent/cert/` directory. (`keystore.jks` is available here).
2. Run the following command:

   ```
   keytool -importcert -keystore keystore.jks -storepass password -alias
   alias_name -noprompt -file certificate_file
   ```

   Where:

   - `-storepass password`: The default, initial password for the agent keystore. Refer to your `keytool` documentation for the default `storepass` password. See keytool.
   - `-alias alias_name`: Any name to uniquely identify the imported certificate in the keystore.
   - `-file certificate_file`: Absolute path of the certificate file.

Run the Connectivity Agent Installer as a Background Process on Linux Systems

When you run the connectivity agent installer (using `java -jar connectivityagent.jar`), the process is tied to the terminal window in which you are working and ends when the window is closed. If you want to run the process in the background, use one of the following options:

- Run the Connectivity Agent Installer as a Background Process
- Connectivity Agent Cannot Be Run as a Service on Windows

Run the Connectivity Agent Installer as a Background Process

1. Update `InstallerProfile.cfg` with `oic_USER` and `oic_PASSWORD` values and then use `nohup` to run the agent process. For example:

   ```
   nohup java -jar connectivityagent.jar &
   ```

Or

1. If you do not want to expose the password in `InstallerProfile.cfg`, perform the following steps:
   a. Enter `java -jar connectivityagent.jar` at the command prompt.
   b. Enter the username and password when prompted.
   c. Enter `Ctrl+Z` to suspend the process.
   d. Enter `bg` (to run the process in the background).
   e. Enter `jobs` to get the `jobid`.
   f. Enter `disown -a %jobid` (from Step e) to disassociate the process from the owning shell.

Connectivity Agent Cannot Be Run as a Service on Windows

Running the connectivity agent as a service on Windows is *not* supported. You must be logged in to the Windows host. Otherwise, the agent stops working when you log off. Consider the following recommended options.

1. Create a new service account on Windows.
2. Use that account to log in to the Windows host and remain logged in with that account.
3. Install the connectivity agent on the Windows host.
4. Use that agent to communicate with the MS SQL Server or other endpoints accessed on Windows.

or

1. Instead of installing the connectivity agent on a Windows host, install it on a Linux virtual machine that has access to the Windows systems with which that agent must interact.

Restart the Agent

You can restart the on-premises connectivity agent if required.

1. Stop the agent in either of the following ways:
   - Enter `ctrl+c` on the host on which the agent is running.
   - Search for the connectivity agent process and kill it.
2. Wait at least 45 seconds before restarting the agent. This is because the agent monitoring framework waits for 45 seconds before marking the agent status as being down. If you start the agent before 45 seconds have completed, agent restart fails with the following error message:

   ```
   Agent is already running for this particular instance
   ```

3. Restart the agent based on your environment and production load size.

Environment	Enter the Following Command

Production environments	It is recommended that you restart the agent with the `-XX:+HeapDumpOnOutOfMemoryError` parameter:
	
	`java -XX:+HeapDumpOnOutOfMemoryError -jar connectivityagent.jar`
	
	If the connectivity agent runs out of memory, this parameter by default ensures that the heap dump is stored in a `java_pidpid.hprof` file in the directory where the agent application is run.
	Based on production loads, it may sometimes be necessary to allocate a larger amount of heap size for the agent process. If you determine that the process must be allocated a larger heap size, tune the `-Xms` and `-Xmx` parameters accordingly.
	• `-Xmsheap_sizeG`
	• `-Xmxheap_sizeG`
	For example, assume you want to assign a minimum of 2 GB and a maximum of 8 GB to the agent JVM.
	
	`-Xms2G -Xmx8G`
	
	Start the agent with the following parameters.
	
	`java -Xms2G -Xmx8G -jar connectivityagent.jar`
	
	The `-XX`, `-Xms`, and `-Xmx` parameters can all be specified at the same time, if needed.

Nonproduction environments	`java -jar connectivityagent.jar`

You can restart the agent as a background process. See Run the Connectivity Agent Installer as a Background Process on Linux Systems.

Create a Connection with an Agent Group

After you have installed the on-premises agent, you can create a connection that uses the agent group and its associated on-premises agent. Only agent groups whose monitoring status is green on the Agent Monitoring page and which have not yet been associated with an adapter can be selected.

To create a connection with an agent group:

1. On the Oracle Integration home page, click **Home** > **Integrations** > **Connections**.
2. Click **Create**.

   See Create a Connection.

3. Select the adapter to configure as a target endpoint. See About Creating Hybrid Integrations Using Oracle Integration .
4. Configure the connection properties and security properties.
5. In the **Agent Group** section, click **Configure Agents** to select the agent group to associate with the adapter. This enables you to access your on-premises applications.
6. Select the agent group to use with this adapter, and click **Use**.
7. Click **Test**. This test executes the ping command on the on-premises instance when the connection is associated with an agent.
8. Click **Save**, then click

.
9. Create an integration in which you use the adapter.
10. Activate the integration. See Activate an Integration.
11. Invoke the integration.

> **Note:**
>
> If you receive the following error, a connection time out has occurred. The request may be slow, in which case the request must be executed again. You can also view the agent logs to see what may be causing the request to not process.
>
> ```
> CASDK-0005 A connector specific exception was raised by the
> application. oracle.cloud.cpi.omcs.api.CpiOmcsException.
> No response received within response time out window of 60000.
> ```

Upgrade the Connectivity Agent Automatically

When a new version of the on-premises connectivity agent becomes available, your host is automatically upgraded with the latest version. When Oracle Integration is upgraded, the agent is upgraded within a four hour window. There is no separate alert for agent upgrade. There is no downtime or interruption of service for in-process integrations that use the agent. If there is any failure for in-process integrations using the agent, those integrations may require resubmission.

Connectivity Agent Upgrade Process

Connectivity agent upgrade occurs as follows:

1. A check is made of the version of the agent installed on your on-premises host.
2. If the agent version on your host is older than the latest available version, the new version is downloaded to your host.
3. The downloaded ZIP file is unzipped.
4. A backup directory is created.
5. A backup copy is made of your existing installation in the new backup directory.
6. Older artifacts are replaced in the `agenthome` directory of your installation.

7. The endpoints are quiesced.
8. The agent is shut down and restarted.
   You are notified of upgrade success.

> **Note:**
>
> After agent upgrade occurs, the agent fails to initialize the SAP JCO libraries because these are native libraries and can only be loaded once per JVM. In the case of automatic agent upgrade, the agent restart reloads the classes (after agent upgrade) without shutting down the JVM. As a workaround, a manual restart of the agent is required. See Restart the Agent.

View Connectivity Agent Restart Progress on Windows

The connectivity agent is automatically restarted after upgrade completes. During the restart on Windows, the connectivity agent runs in the background.

To view connectivity agent restart progress, you can look in the PID file (`oic_connectivity_agent/pid`) for the process ID, run the `jps` command from the prompt (as shown below), or view the connectivity agent upgrade logs located in `agenthome/logs/agent-diagnostic0.log` and `agenthome/logs/agent-upgrade.log`.

```
C:\OIC\Agent\Agentwindows\ADT\oic_connectivity_agent>jps
12504 OracleIdeLauncher
13944
17960 connectivityagent.jar
19496 Jps
4104 OracleIdeLauncher
```

Back Up and Recover the Connectivity Agent

If the host on which the connectivity agent is running crashes and makes your environment unavailable, follow these steps to ensure that the connectivity agent is recovered and functions correctly.

> **Note:**
>
> - Perform regular backups of the `agent` directory located under *`AGENT_INSTALL_DIR`*`/agenthome`.
> - Ensure that the regular backups are not on the same physical host on which the agent is currently running.
> - Ensure that backups are available in the event of an agent host crash.

If your host crashes and makes the environment unavailable, do the following;

1. From the left navigation pane, click **Home** > **Integrations** > **Agents**.
2. Click **Download** > **Connectivity Agent**.

3. Save the `oic_connectivity_agent.zip` file on the agent host.
4. Create an *`AGENT_INSTALL_DIR`* directory. The new *`AGENT_INSTALL_DIR`* directory name *must* match the old name.
5. Unzip `oic_connectivity_agent.zip` under the *`AGENT_INSTALL_DIR`* directory.
6. Replace the new `agent` directory in *`AGENT_INSTALL_DIR`*`/agenthome` with the `agent` directory from your last backup.
7. Restart the connectivity agent.

```
java -jar connectivityagent.jar
```

> **Note:**
>
> If you use third party JAR files, ensure that the same JARs are available under *`AGENT_INSTALL_DIR`*`/agenthome/thirdparty/lib`. If not, add them to the `lib` directory.

> **Note:**
>
> - To ensure the integrity of the backups, always create a new backup if integration activations/deactivations have occurred since the last backup.
> - The agent instance name is from the backed up configuration and is mapped to the current host. If the backup and recovery is performed on a different host, the agent instance name is not changed to reflect the new host on which the agent is running.

Use the Connectivity Agent in High Availability Environments

You can use the connectivity agent in high availability environments with Oracle Integration. You install the connectivity agent twice on different hosts. The connectivity agents can scale horizontally, thereby providing all the benefits of running multiple agents for an agent group. This results in increased performance and extends failover benefits.

- Prerequisites
- Install the Connectivity Agents
- Features and Considerations

Prerequisites

Ensure that both agent instances can access the same endpoints. For example, agent 1 on host 1 and agent 2 on host 2 must both be able to access the same endpoint (for example, a Siebel system).

Install the Connectivity Agents

1. Create a new agent group. You cannot use an existing group.
2. Download and install the first agent on one host. Ensure that you specify the agent group identifier of the new agent group.

See Download and Install the Agent.

3. Follow the same steps to download and install the second agent on a second host. That way, if one host goes down, agent processing continues running on the other host.
   - Use the same JAR file that you downloaded for the first installation.
   - Use the same `InstallerProfile.cfg` file settings. You can also copy the file from one host to the other.

   > **Note:**
   >
   > Specify the *same* agent group identifier and Oracle Integration URL in the `InstallerProfile.cfg` file as with the first installation.

4. After installation completes, go to the Agents page and note that two agents are associated with the same agent group.
5. Go to the Monitor Agents page and note the same information. In this case, both agent instances are running.

Features and Considerations

- You can install multiple agents on the same host. However, to utilize high availability capabilities, install the second agent on a second host.
  - Create a horizontal cluster to achieve high availability. Installing the agent on the same virtual machine (VM) does not guarantee high availability.
- You cannot have more than two agent instances per agent group. Attempting to include a third agent instance in the same group during installation results in an error.
- While you can install up to two agent instances per agent group, be aware of the current restriction with agent high availability when working in tandem with automatically upgraded agents. To ensure that both agent instances do not upgrade at the exact same time, it is recommended that you start each instance with a small time gap. The time gap can be as little as 10 minutes. This ensures that there is no outage due to both agent instances being upgraded at the same time.
- The File Adapter and FTP Adapter are not supported in high availability environments. When using the File Adapter or FTP Adapter and some groups have multiple instances, use a dedicated agent group (with one agent only).

Delete an Agent Group

You can delete an agent group that is not currently associated with a running connectivity agent instance. Deleting the agent group also deletes the internal artifacts that the agent relies on when the agent group was created.

To delete an agent group:

1. Before deleting an agent group, ensure that you kill the running agent at the operating system command prompt:

   ```
   kill -9 agent_PID_number
   ```

   The agent_PID number can be obtained from the *AGENT_INSTALL_DIRECTORY*`/pid` file.
2. In the left navigation pane, click **Home** > **Integrations** > **Agents**.

3. Find the agent group to delete. If the number count for the agent group is zero, you can delete the agent. Otherwise, you must first click the number (it can be 1 or 2). Delete those agent instances first. If any connections are using the agent, you cannot delete the agent instance.
4. Select **Delete** from the  menu.
5. Select **Delete** when prompted to confirm.

Agent Behavior in a Decommissioned Instance or HTTP 404/401 Error Response Codes

The connectivity agent continuously seeks work to process. For cases in which the Oracle Integration instance is decommissioned or if all requests to the instance from the agent fail with an HTTP error code of 401/404 for a continuous period of 24 hours, the agent terminates the poller threads. This halts all message processing. When the conditions leading to the error have been resolved, the agent must be restarted manually.

See Restart the Agent.

9

Use JavaScript Libraries in Integrations

You can add JavaScript libraries into integrations.

Topics:

Use Libraries to Manage Integration Functions

JavaScript libraries provide a means to create and organize JavaScript for use in integrations. Library functions are automatically available for you to drag into your orchestration integrations.

You can use the library functionality to:

- Create a new JavaScript library
- See a list of all created JavaScript libraries
- View the details of a JavaScript library
- Edit the details of a JavaScript library
- Determine which integrations are using library functions
- Delete a JavaScript library
- Import functions
- Download a library source file to your desktop
- Add library functions in an orchestrated integration

JavaScript libraries are not provided out of the box. You can create your own. However, the Expression Builder and the mapper automatically provide built-in functions. See Add Functions, Operators, and XSLT Statements in *Using the Oracle Mapper with Oracle Integration Generation 2*.

Create a JavaScript Library

In order for custom function calls to appear in the integration **Actions** palette, you must create the libraries that contain them.

1. In the left navigation pane, click **Home** > **Integrations** > **Libraries**.
2. Click **Create** in the banner.
3. In the Create Library dialog box, click **Browse** to select a JavaScript (`.js`) library file.
4. Specify a name, identifier, version number and optional description.

Example 9-1 Code Format

Function return parameters must be named like the example below. You should configure the return parameter type in the metadata user interface only if the return parameters are named.

Consider the following code:

```
function add ( param1, param2 )
 {    return param1 + param2;
}
```

Although the above syntax is valid, the return parameter does *not* appear in the metadata user interface for configuration. For that to happen, the code must return a named parameter. Create a new variable and assign return values before the variable is returned. For example, like this:

```
function add ( param1, param2 ) {
   var retValue = param1 + param2;
   return retValue;
}
```

Example 9-2 Function Within Another Function

Functions defined within another function do not appear in the metadata user interface for configuration.

Consider the following code:

```
numeric.parseDate = function parseDate(d) {
    function foo(d) {
        if(typeof d === 'string') { return Date.parse(d.replace(/-/g,'/'));}
     if(!(d instanceof Array)) { throw new Error("parseDate: parameter must
be arrays of strings"); }
     var ret = [],k;
     for(k=0;k<d.length;k++) { ret[k] = foo(d[k]); }
        return ret;
    }
     var retVal = foo(d);
     return retVal;
   }
```

In this case `foo` is defined within `parseDate`. Therefore, `foo` does not appear in the metadata user interface for configuration.

View Library Details

View details about a library without changing any information.

1. In the left navigation pane, click **Home** > **Integrations** > **Libraries**.
2. Go to the row of the library to view.
3. Click
   
   .

   The files and their constituent functions are displayed. You can expand and contract the list of functions and you can search and filter the list as well. To view information about a function, click on the function. To view the source code, click on the file name and download the code to your desktop. You can also click on the file name to annotate the entry with your comments.
4. Use the
   
   menu on this page to view primary information about the library and to export XML metadata to your desktop.
5. When done, click
   
   to return to the Libraries page.

Edit Library Details

Edit details about a library.

1. In the left navigation pane, click **Home** > **Integrations** > **Libraries**.
2. Either click the library name or click  .

   The files contained in the library and their constituent functions are displayed. You can expand and contract the list of functions, and you can search and filter the list. To edit information about a function, click on the function. To view the source code, click on the file name and download the code to your desktop. You can also click on the file name to annotate the entry with your comments.

   If the function is in use in an integration, you are warned about it and the function is presented in view-only mode.

   If the function is available for editing, you can set the data types of the function's input and output values, defining them as **Boolean**, **Number** or **String**. You can also add a description.

   You can use the  menu on this page to:

   - View primary information about the library
   - Delete the library
3. When done, click **Save** in the banner.

Determine Which Integrations Use a Library Function

You can determine which integrations are using functions in a given library.

1. In the navigation pane, click **Home** > **Integrations** > **Libraries**.
2. Go to the row of the library to check.
3. In the **Usage** column, click the number. This value indicates the number of integrations that use functions in this library.

   When you click the number, the Library Is In Use dialog is displayed. It lists the names of the integrations that use the library's functions. This dialog is also invoked if you attempt to edit or delete a library that is currently in use.

> **Note:**
>
> If you attempt to activate an integration that includes a function that is not completely configured, an error message is displayed in the banner. You must complete configuration of this function before you can activate the integration. See Activate an Integration.

Delete a Library

Delete a library.

1. In the left navigation pane, click **Home** > **Integrations** > **Libraries**.
2. Go to the row of the library to delete.
3. Select **Delete** from the

   menu.

   If any of the library's functions are being used by an integration, the Library Is In Use dialog is displayed and you are prevented from deleting the library. You must remove the functions from the integration before you can delete the library.

Import a Library File

You can import a library ZIP file into Oracle Integration. The ZIP file must contain only the library JAR and metadata XML file.

1. In the left navigation pane, click **Home** > **Integrations** > **Libraries**.
2. Click **Import** in the banner.
3. In the Import Library dialog box, click **Browse** to select the ZIP file.

   The Javascript Library page is displayed.
4. Make updates, as necessary, then click **Save**.

Download a Library File

You can download a library to a source file (`.jar` or `.js`) on your desktop.

1. In the navigation pane, click **Home** > **Integrations** > **Libraries**.
2. Go to the row of the library to download.
3. Select **Download** from the  menu.
4. Click **Save File** to save the file to your desktop.

Update Library Functions

You can update an already created library with an updated Javascript file.

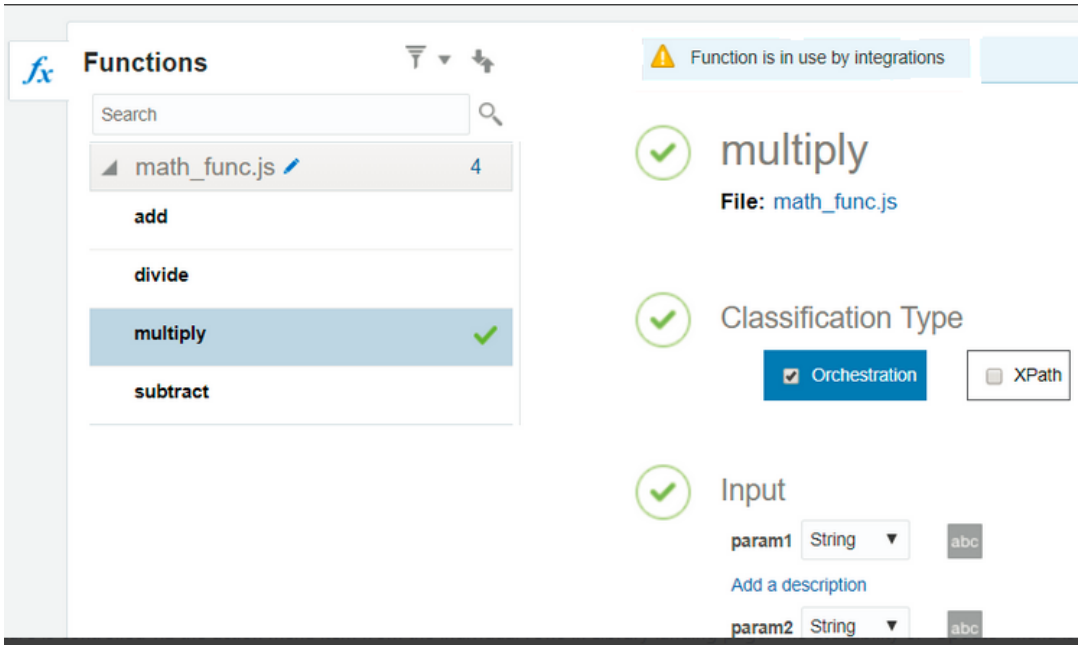
Note the following guidelines for updating a Javascript file:

- Functions (within a library) used in integrations must be available in the updated Javascript file.
- The signatures (parameters and function name) for these functions must remain intact to prevent runtime issues.
- The addition of new functions is allowed.
- Active integrations using new or updated functions must be reactivated.

The following changes make an updated Javascript file invalid:

- Changing the name of a function being used in an integration.
- Changing the parameter definitions of functions used in an integration.
- Structural issues with the Javascript file.
- The number of functions exceeds 500.

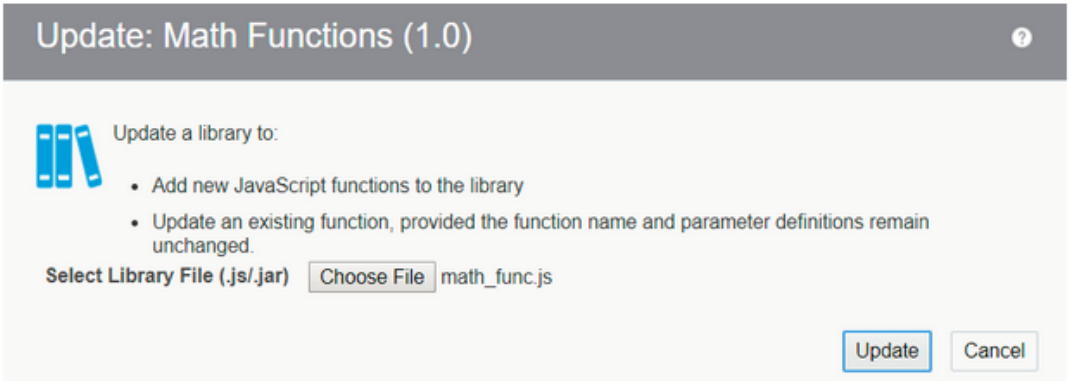
1. In the left navigation pane, click **Home** > **Integrations** > **Libraries**.
2. Find the library to update.
3. Select **Update** from the menu. For example, assume the library to update includes four functions, one of which (`multiply`) is currently being used in an integration.



The content of the library is as follows:

```
function add ( param1, param2 ) {
     var retValue = param1 + param2;
     return retValue;
}
function subtract ( param1, param2 ) {
     var retValue = param1 - param2;
     return retValue;
}
function multiply ( param1, param2 ) {
     var retValue = param1 * param2;
     return retValue;
}
function divide ( param1, param2 ) {
     var retValue = param1 / param2;
     return retValue;
}
```

4. Click **Choose File** to select the Javascript file that includes the function updates, then click **Update**.

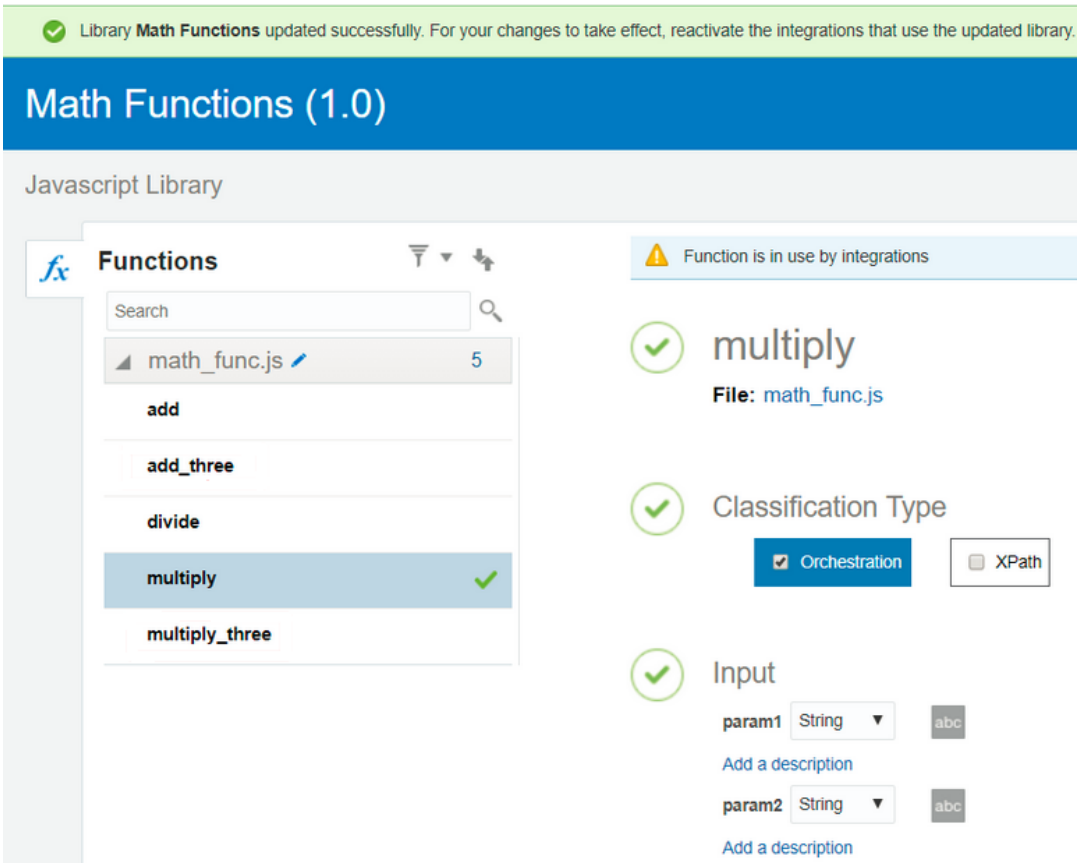


For this example, the Javascript file consists of the following updates:

- Two new functions: `add_three`, `multiply_three`
- One removed function that was not being used by any integrations: `subtract`
- A code change to the `multiply` function that is currently being used in an integration

```
function add ( param1, param2 ) {
     var retValue = param1 + param2;
     return retValue;
}
function multiply ( param1, param2 ) {
     var retValue = param1 * param2 * 2;
     return retValue;
}
function divide ( param1, param2 ) {
     var retValue = param1 / param2;
     return retValue;
}
function add_three ( param1, param2, param3 ) {
     var retValue = param1 + param2 + param3;
     return retValue;
}
function multiply_three ( param1, param2, param3 ) {
     var retValue = param1 * param2 * param3;
     return retValue;
}
```

All these updates are valid and the library is successfully updated. The library edit page is displayed with the updates. Any active integrations using the `multiply` function must be reactivated.



An example of an invalid Javascript file that would *not* be successfully uploaded is if the `multiply` function included a parameter definition that changed (for example, `param2` to `param3`).

```
function add ( param1, param2 ) {
     var retValue = param1 + param2;
     return retValue;
}
function multiply ( param1, param3 ) {
     var retValue = param1 * param2 * 2;
     return retValue;
}
function divide ( param1, param3 ) {
     var retValue = param1 / param2;
     return retValue;
}
```

Use XPath Functions in JavaScript Libraries

You can use the following XPath functions in JavaScript libraries in Oracle Integration.

- ```
  function sha256(ascii) {
      result = xpath.ora.checksum(ascii, "sha-256");
      return result;
  }
  ```
  ```

- ```
  function sha512(ascii) {
      result = xpath.ora.checksum(ascii, "sha-512");
      return result;
  }
  ```

- ```
  function sha(ascii) {
      result = xpath.ora.checksum(ascii, "sha");
      return result;
  }
  ```

- ```
  function md5(ascii) {
      result = xpath.ora.checksum(ascii, "md5");
      return result;
  }
  ```

- ```
  function encode(input){
      result = xpath.ora.encodeData(input, "b64mime");
      return result;
  }
  ```

- ```
  function decode(input){
      result = xpath.ora.decodeData(input, "b64mime");
      return result;
  }
  ```

Use Library Functions in Orchestrated Integrations

You can add library functions to orchestrated integrations.

After you create and edit your library functions, they are available from the  orchestration palette or the  icon inside an integration. See Add a JavaScript Action.
  ```

10

Integrate with Processes and Services

You can integrate Oracle Integration with processes and other services.

Topics:

Create an Integration that Invokes a Process

You can invoke a process from an orchestrated integration. When you drag the process node into an integration, the Select Process wizard is invoked and prompts you to select an application workspace and a process to be invoked.

To invoke a process from an orchestrated integration:

> **Note:**
>
> When calling a process from an integration, use OAuth-based authentication. Client ID/secret-based authentication is not supported for integration-to-process calls.

1. In the left navigation pane, click **Home** > **Integrations** > **Integrations**.
2. In the upper right, click **Create**.
3. Select **App Driven Orchestration**.

   The Create New Integration dialog is displayed.
4. Complete the fields to create an orchestrated integration, then click **Create**.
5. Add a process action to an integration in either of the following ways:
   - On the right side of the canvas, click 
     and drag the **Process** action to the appropriate location.
   - Click  at the location where you want to add the process action, then select **Process**.

   The Process Selection wizard is displayed.
6. On the Basic Info page, enter a name and optional description, then click **Next**.
7. On the Select Process page, enter the following details:
   a. In the **Application** list, select the application workspace. The available application workspaces are displayed for the process instance.

b. In the **Process (Message Based Pattern)** field, select the process name. The process must already be activated and must be an asynchronous, message-based process to be visible for selection.

c. If your process has multiple operations, select an operation from the **Operation** list. Processes typically have only a single operation that is selected by default.

d. To use the default version of the process, select the **Use Default Process Version** check box. Selecting this check box causes the default process version to be triggered regardless of the version you selected.

   For example, assume the process you selected in the **Process (Message Based Pattern)** field is using process version 4.0, and the process version is later upgraded to version 4.1. By selecting this check box, your integration doesn't need to be modified to call the latest default process version. The integration automatically starts using the latest default version of the process (for this example, version 4.1).

e. Click **Next**.

8. On the Summary page, confirm your selections and click **Done**.

   A process node with a map is displayed to provide the inputs to invoke the process. The map may contain one or more schemas and one or more complex or primitive types depending on the interface defined for the process.

9. Complete the design and save the integration.

10. Activate and invoke the integration.

    Every time a process is invoked, a new instance of the process is created.

Connect to Oracle Mobile Cloud Service

Oracle Mobile Cloud Service (MCS) enables you to create an Oracle Integration Connector API to connect to Oracle Integration, which in turn, enables you to connect to, browse, and select services that are defined in on-premises applications and other cloud services.

See *Using Oracle Mobile Cloud Service*.

Connect to Oracle API Platform Cloud Service

Connect to Oracle API Platform Cloud Service to manage your integrations as APIs and publish them to the API Platform Cloud Service Developer Portal for consumption.

Before you begin: Ensure that you have a subscription to Oracle API Platform Cloud Service. You must also have credentials for a user assigned the Administrator or API Manager role in API Platform Cloud Service.

To connect to an API Platform Cloud Service instance:

1. In the left navigation pane, click **Home** > **Settings** > **Integrations** > **API Management**.

2. Click **API Platform**.

3. In the **Connection Name** field, enter a name for the connection.

4. In the **URL** field, enter the URL for the API Platform Cloud Service Management Portal instance to use with Oracle Integration following this pattern:

   ```
   https://hostname_or_IP:port
   ```

5. In the **Username** and **Password** fields, enter the credentials for your API Platform Cloud Service user. This user must be issued the Administrator or API Manager role in API Platform Cloud Service.
6. Click **Save**.

If the URL and credentials you entered are valid, the connection is saved and a confirmation message is displayed.
See Manage an Integration as an API with Oracle API Platform Cloud Service to manage your integrations as APIs with API Platform Cloud Service.

Invoke a Co-located Integration from a Parent Integration

You can invoke a co-located (child), active integration from the parent integration that you are designing through use of the local integration adapter. Co-located means the integration is running on the same host instance or in the same domain. Upon activation and invocation of the parent integration, it invokes and consumes the co-located integration.

- Capabilities
- Restrictions
- Best Practices for Designing Local Invoke Calls
- Invoke a Co-located Integration with a Parent Integration

Capabilities

The local integration adapter is a system adapter that is not exposed on the Connections page when you create a connection. Instead, the local integration adapter is invoked by a preseeded connection. It is created when creating an integration, if the local integration connection does not exist. This adapter has no connection properties and no security policies.

The local integration adapter works as follows:

- You select an active integration from runtime (that is, co-located) to build a new integration. Note the following guidelines:
  - Active synchronous-synchronous, asynchronous with no callback, and scheduled integrations are available for selection.
  - Asynchronous with callback integrations do not appear.
  - De-activated and draft integrations do not appear.
  - Scheduled integrations are permitted as an invoke connection and automatically called by the **Submit Now** option.
- You can add multiple child integrations. You can also add an integration containing a child integration (embedding multiple levels).
- You can map return values downstream. This mapping functionality is the same as for any other invoke mappings in the integration.
- You can add header support to the payload of the SOAP Adapter or REST Adapter used in the co-located (child) integration. See Create a Co-located Integration with Header Support.
- You can create a parent integration using a REST Adapter invoke connection configured with the OpenAPI URL connection type to pass a binary payload to a child integration. Use binary data with payloads that are unstructured and inline (for example, application/octet-stream). The file contents are preserved, but require the receiver to determine the file

type (for example, from the file name extension). The internet media type for an arbitrary byte stream is application/octet-stream.

- You can dynamically invoke child integrations from a parent integration at runtime. This feature is useful for scenarios in which you must:
  - Invoke a REST endpoint dynamically or when the endpoint is not known at design time.
  - Invoke multiple REST child integration endpoints, all of which accept the same input payload and return the same response payload. This feature eliminates the need to design all child integrations in a switch action.

  See Dynamically Invoke a Co-located Integration.

The local integration adapter and the existing SOAP Adapter and REST Adapter differ as follows:

- You do not need to create multiple connections (SOAP/REST) to invoke the local integration service.
- Runtime communication always uses HTTP (that is, non-SSL).

This adapter is invoked internally in Oracle Integration to perform the local service invocation.

Restrictions

Note the following local integration restrictions:

- You cannot send or receive file references between parent and child integrations, in either direction, whether using local integrations or otherwise. The stage file is scoped in the context of its own flow instance and is not shared between parent and child integrations. This is by design. Instead, you can send or receive the stage file as an attachment instead of a direct reference.
- OAuth with client credentials is not recommended.
- You cannot create a child integration with a WSDL having a nested anonymous schema. As a workaround, manually edit the WSDL and make the schema non-anonymous.
- SOAP-based child integrations:
  - A parent integration cannot send an attachment to a configured child integration.
- REST-based child integrations:
  - A parent integration cannot send an attachment to a configured child integration.
  - A parent integration cannot process an XML payload that is expected by the child integration. The child flow must be configured to receive a JSON payload only, and cannot be invoked locally using an XML payload.

Best Practices for Designing Local Invoke Calls

- Calls to schedule tasks or asynchronous integrations are always fine with a local invoke because they are not executed in the context of the current node.
- Calls from nonparallel code sections are also always fine because the calling integration block is waiting for a response. There is currently no parallel capability in synchronous integrations. Therefore, a local invoke is always fine in a synchronous flow.
- Calls from parallel code sections (parallel for-each and parallel file processing) are problematic when using a local invoke because they may use the same node for execution. In those cases, calls do not benefit from parallelism across nodes and instead are competing for resources on a single node.

Invoke a Co-located Integration with a Parent Integration

The following steps provide an overview of creating an orchestrated integration in which a parent integration invokes a co-located integration.

> **Note:**
>
> - If you need to move a parent integration to another instance, it is best to include that integration and all its child integrations in a package.
> - If the following integrations are imported from one environment to another (having different host names), then editing the local (child) integration or editing the REST Adapter in the Adapter Endpoint Configuration Wizard leads to major changes in the mapper that may require remapping.
>   - Integrations in which a co-located (child) integration is invoked from a parent integration.
>   - Integrations with a REST adapter using a Swagger-based connection.

1. Create and design an orchestrated integration.
2. Add an integration action to an integration in either of the following ways:
   - On the right side of the canvas, click 
     and drag the **Integration** action to the appropriate location.
   - Click  at the location where you want to add the integration action, then select **Integration**.

   The Configure Local Integration Selection Wizard is displayed.
3. Specify the following details, and click **Next**.

Element	Description
What do you want to call your local integration invocation?	Specify a name.
What does this local integration invocation do?	Specify a description.

4. Specify the following details, and click **Next**.

Element	Description
Integration	Select the co-located integration to invoke. Only active integrations are displayed.
Identifier	Displays the identifier of the selected integration.
Description	Displays the description of the selected integration.

5. Select the operation for the co-located integration to perform, and click **Next**. The operation depends on the child integration. Note the following:

- If the child integration trigger connection is a REST Adapter, it can have Get, Post, Put, and Delete operations.
- If the child integration trigger connection is a SOAP Adapter, it can have the WSDL operation name.
- If the child integration trigger connection is a schedule, it executes a submit now configuration.

6. Review your selections, and click **Done**.
7. Complete design of the parent integration.
8. Specify appropriate business identifiers for tracking the integration at runtime.
9. Activate and invoke the integration.
10. Go to the Track Instances page to track integration status.
    If successful, the Track Instances page has two instances (one for the parent and one for the child). If unsuccessful, and the parent is unable to call the child, there is one entry (for the parent). If the parent can call the child, there are two entries. If the child fails or the child was successful, but the parent the did not process properly the response, the parent may fail.

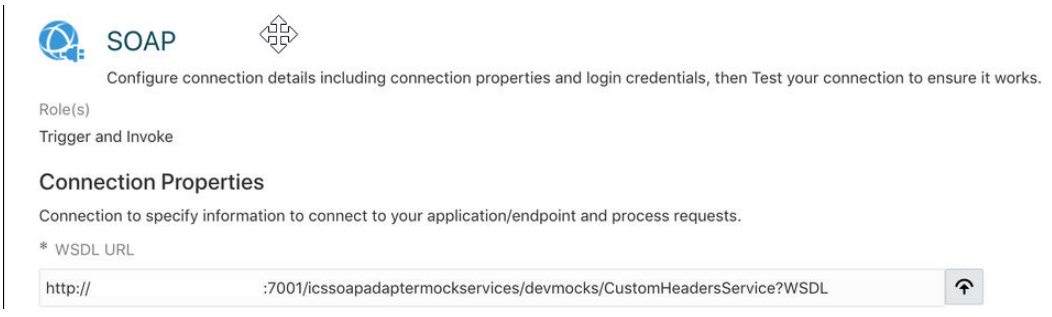
Create a Co-located Integration with Header Support

Headers are optional elements that pass extra information about your application requirements. For example, you can use the header element to specify a digital signature for password-protected services. You can configure and select the headers to send with the payload. Headers selection is automatic based on the WSDL URL definition you provide on the Connections page.

Along with request and response SOAP and REST headers, you can configure standard HTTP headers with the SOAP Adapter and REST Adapter, custom HTTP headers with the SOAP Adapter and REST Adapter, and custom SOAP headers. You can view and edit these headers in the request and response mapper under the header elements for either the SOAP Adapter or REST Adapter.

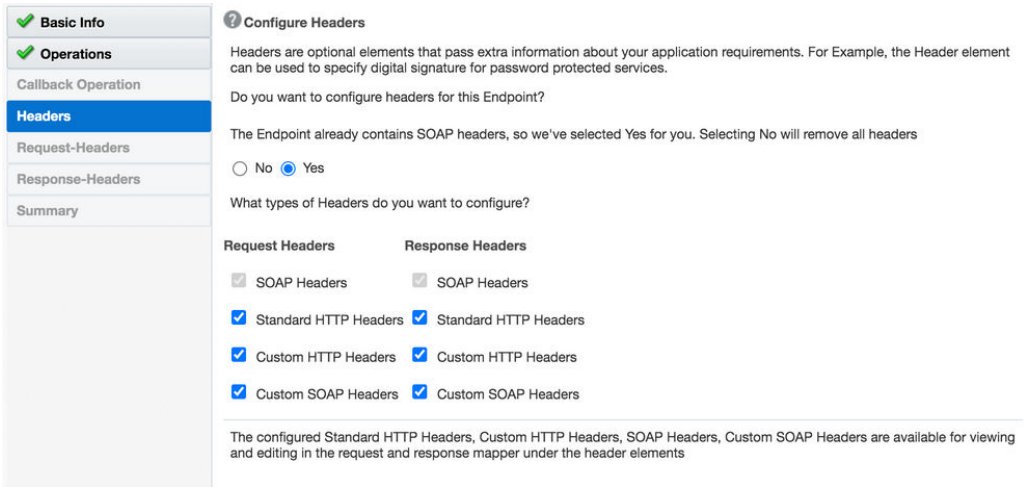
The following use case provides a high level overview of creating a co-located integration with header support.

1. Create a SOAP Adapter connection (for this example) with a WSDL URL that includes the header definitions.

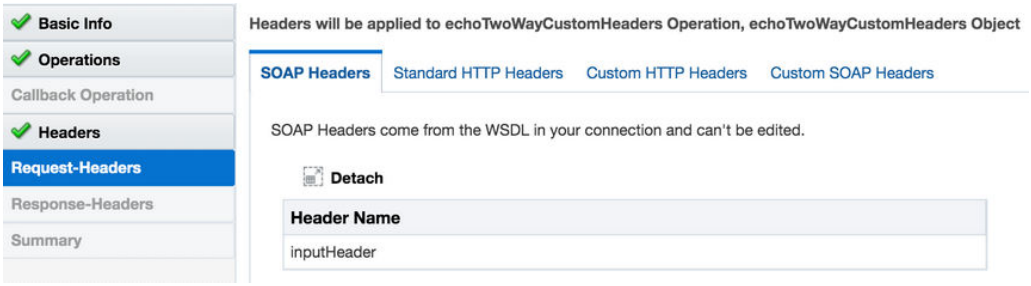


2. Complete the remaining fields on the Connections page.
3. Test and save the connection.
4. Create an orchestrated integration to act as the co-located (child) integration.

5. Add the SOAP Adapter as a trigger connection in the integration.
6. Complete the pages of the Adapter Endpoint Configuration Wizard. In particular:
   a. On the Headers page, specify request and response header details.

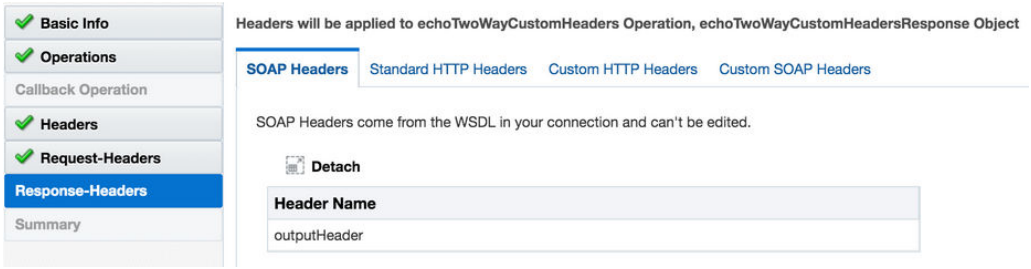


   b. On the Request-Headers page, specify request header names.



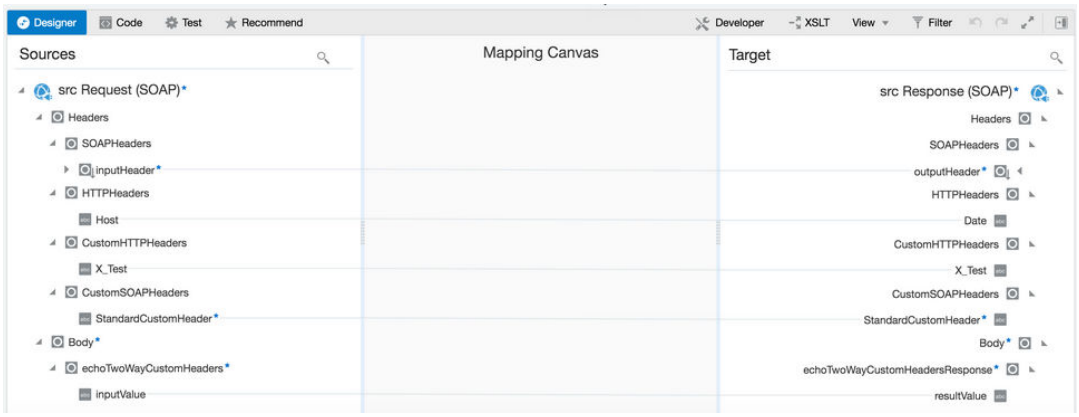
   .

   c. On the Response-Headers page, specify response header names.

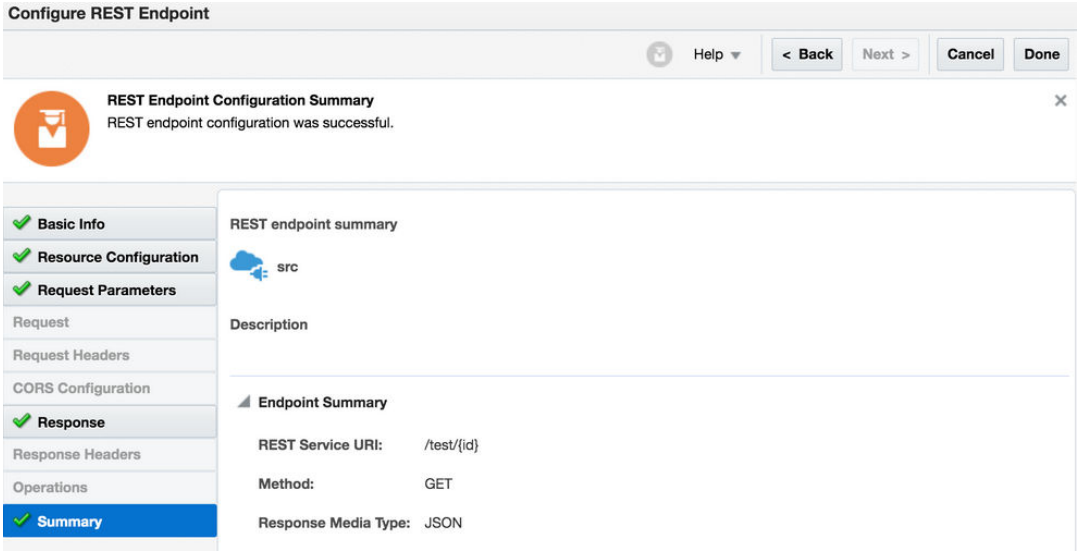


   .

7. In the mapper, map the request and response headers. While this example maps headers between two SOAP Adapters, you can also map request and response headers between two REST Adapters.

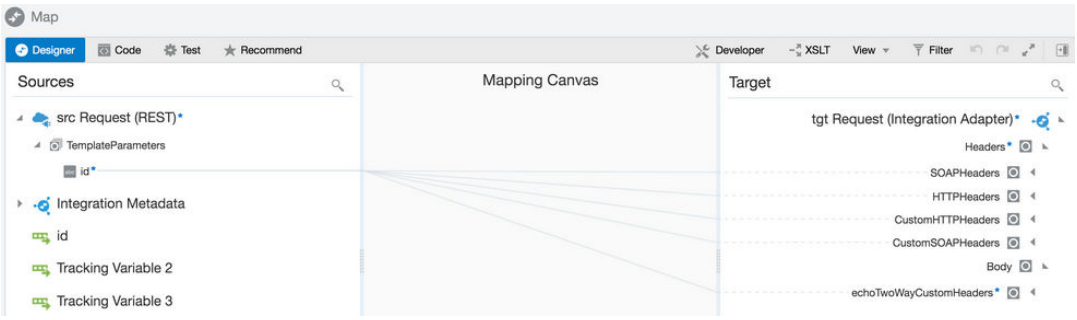


8. Save and activate the integration.
9. Create another orchestrated integration to act as the parent integration.
10. Add the REST Adapter as a trigger connection in the integration.
11. Configure the pages in the Adapter Endpoint Configuration Wizard.
12. On the Summary page, review your selections, then click **Done**.



13. Add an **Integration** action to the integration.

    This invokes the Configure Local Integration Selection Wizard.
14. Complete the pages of the wizard. In particular, select the child integration created in Step 4 on the Selection Integration page.
15. Click the map icon between the REST Adapter trigger connection and SOAP invoke (the **Integration** action you configured) to configure the map.
16. Map the source **id** element to the target SOAP, HTTP, and custom header elements.



17. Save and activate the parent integration. The parent integration is now designed with a child integration as a SOAP invoke.

Backward Compatibility for Header Support

Note the following details when using header support in existing integrations created prior to the release of support for headers in co-located integrations.

Question	Answer
What happens if an existing integration created with a header is imported into an environment in which header support in co-located integrations is supported.	You can activate the integration without any issues. However, if you want to use header functionality, you must edit the adapter by clicking through the pages of the Adapter Endpoint Configuration Wizard, then clicking **Done** on the Summary page. In this scenario, mapping is not deleted; you must to correct it.
What happens if you perform the following steps. 1. Create a child flow with headers in a trigger connection. 2. Use this child endpoint as a SOAP connection. 3. Use the above SOAP connection as a trigger and create an integration in which you set **Do you want to configure headers for this endpoint?** to **No** on the Headers page of the Adapter Endpoint Configuration Wizard. 4. Activate the integration. 5. Invoke this child integration in a parent integration that uses the **Integration** action. 6. Check if the integration generates a wrapper in the local invoke.	Based on the selection of **Yes** or **No**, headers are visible or invisible in the mapping.

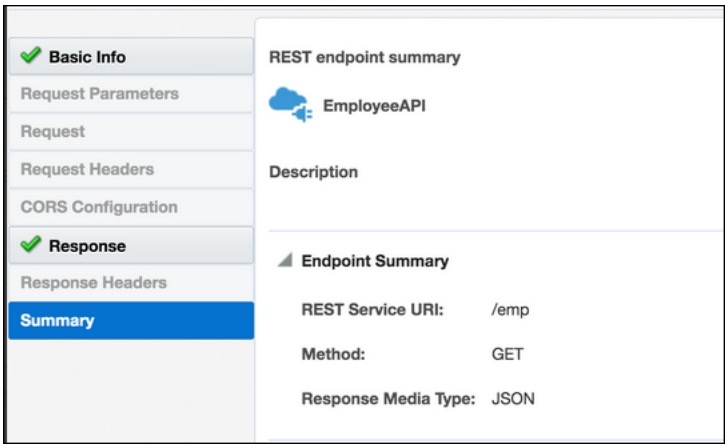
Dynamically Invoke a Co-located Integration

You can dynamically invoke a co-located integration at runtime. In this example, a parent integration can call either of two child integrations at runtime based on the data. The parent integration is designed to invoke child integration one. However, it can also invoke child integration two. The code and version values are passed to the appropriate child integration.

> **Note:**
>
> Dynamic invocations are only supported with the REST Adapter.

1. Create and activate an initial child integration with a REST Adapter trigger connection.
   a. Create an app-driven orchestration integration (for this example, named `childone`).
   b. Add a REST Adapter trigger connection.

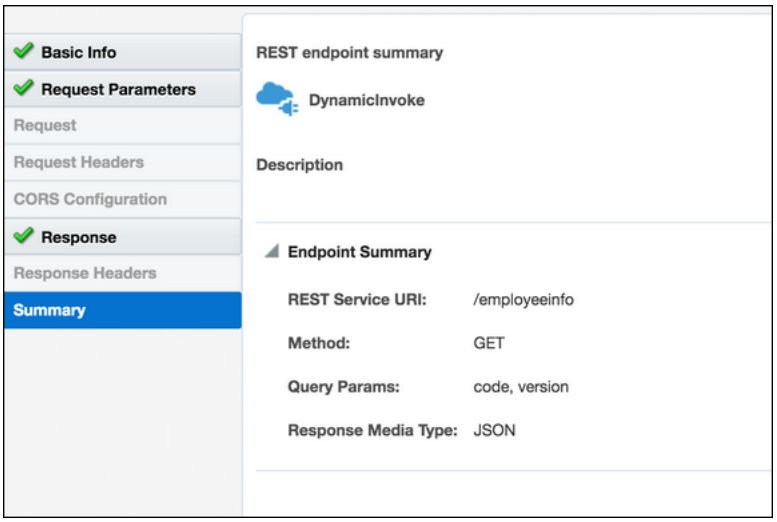
      This invokes the Adapter Endpoint Configuration Wizard.
   c. Provide the necessary details on the pages of the wizard.
   d. Review your configurations on the Summary page, then click **Done**. For this example, the following values are configured:
      - The relative resource URI is **/emp**.
      - The verb (method) is **GET**.
      - The response media is **JSON**.



   e. Perform mappings in the mapper.
   f. Activate the integration.
2. Create and activate a second child integration with a REST Adapter trigger connection.
   a. Create a second app-driven orchestration integration (for this example, named `childtwo`).
   b. Add a second REST Adapter trigger connection.
   c. Configure the second REST Adapter with the same values as the first REST Adapter.
   d. Perform mappings in the mapper.
   e. Activate the integration.
3. Create and activate a parent integration.
   a. Create an app-driven orchestration integration.
   b. Add a REST Adapter trigger connection.
   c. Provide the necessary details on the pages of the wizard, including creating the `code` and `version` query parameters on the Request Parameters page.

d. Review your configurations on the Summary page, then click **Done**. For this example, the following values are configured:

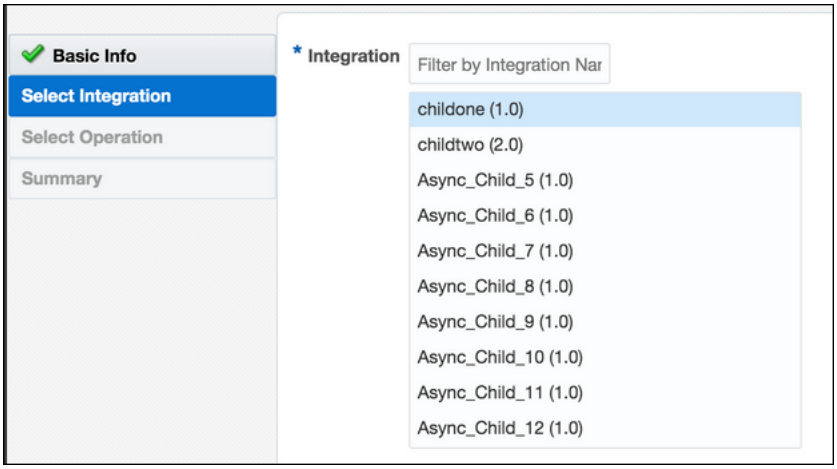
- The relative resource URI is **/employeeinfo**.
- The verb (method) is **GET**.
- The query parameters are **code** and **version**.
- The response media is **JSON**.



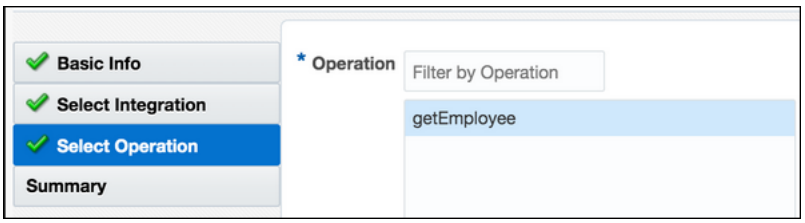
e. Add an **Integration** action to the integration.

This invokes the Integration Adapter Wizard.

f. Select the child integration to invoke. For this example, **childone** is selected.

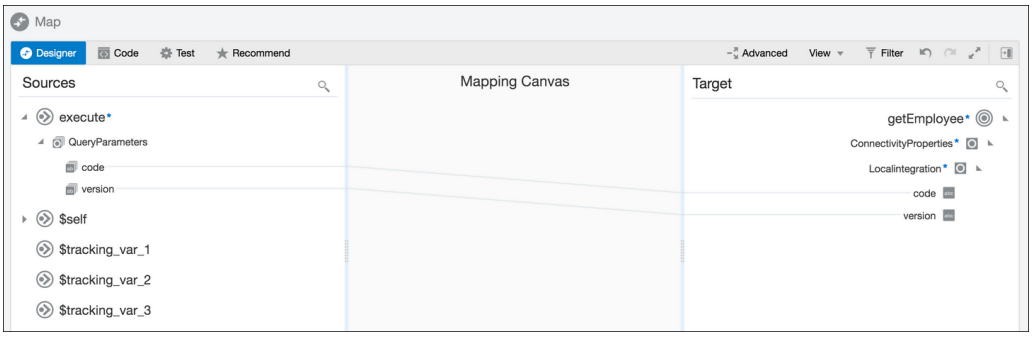


g. Select the operation.



h. Complete the wizard, and click **Done**.

i. Map the **code** and **version** source query parameters to the **code** and **version** target local integration elements.

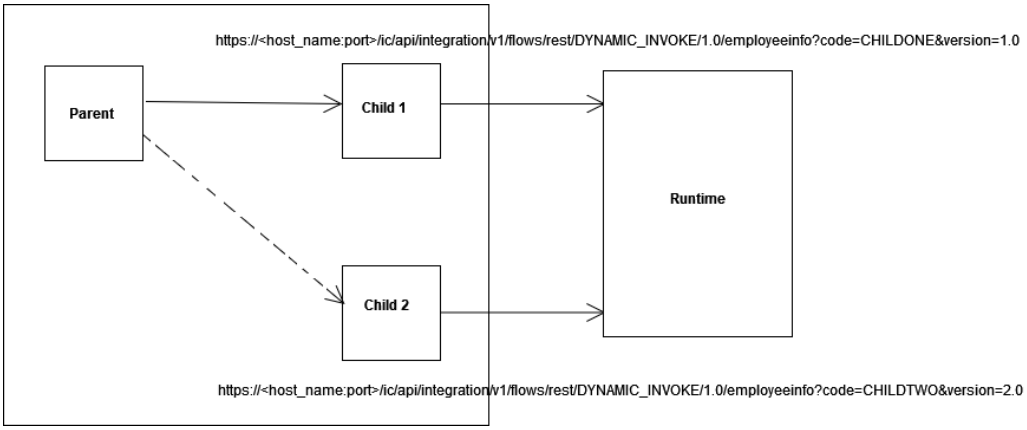


j. Complete design of any remaining actions in the integration.

k. Activate the integration.

Even though the parent integration was designed to invoke only child integration one, you can now invoke either child integration by replacing the code and version of that child integration in the endpoint below.

```
http://host_name:port/ic/api/integration/v1/flows/rest/
DYNAMIC_INVOKE/1.0/employeeinfo?
code=[code-value]&version=[version-value]
```



11

Manage Integrations

When you are ready for your integration to go live, you must activate the integration in Oracle Integration. You can also deactivate a running activation if you must make changes to it or if it is no longer needed. You can perform additional integration management tasks.

Topics:

Activate and Deactivate Integrations

After integration design is complete, you can activate your integration. You can also deactivate integrations.

Topics:

Activate an Integration

Once you create an integration and the progress indicator shows 100%, you can activate that integration to the runtime environment. An integration shows as 100% and is eligible for activation after you have specified the source connection, the target connection, the data mappings, and the tracking fields.

Understand Restrictions About the Number of Active Integrations

The number of active integrations per Oracle Integration instance cannot exceed 700. Active integrations are defined as currently active integrations and integrations whose activations are in progress.

- If you reach 90% of this limit, the following warning is displayed:

  ```
  You've number integrations which are either active or whose activation is
  in progress.
  It is more than 90% of the allowed limit.
  ```

- If you reach the limit of 700, the **Activate** or **Activate & Schedule** button to activate an integration is disabled and the following warning is displayed.

  ```
  You've reached the limit as there are 700 integrations which are either
  active
  or whose activation is in progress. Deactivate or abort the activation of
  an integration
  and try again.
  ```

  If you are nearing your limit, review and delete any older integrations that are no longer required.

To activate an integration:

> **Note:**
>
> If you activate a new version of an existing integration, tracking instances or logs of the old version are not deleted. However, related artifacts are deleted and redeployment is performed on the back end. Monitoring data is also removed.

1. In the left navigation pane, click **Home** > **Integrations** > **Integrations**.
2. Go to the row of the integration to activate.
3. Click the

icon to activate the integration.

   The Confirmation dialog is displayed.
4. Select options appropriate to your integration. A warning message is displayed in the dialog if the number of activated integrations for this Oracle Integration instance is above 90% of the limit.

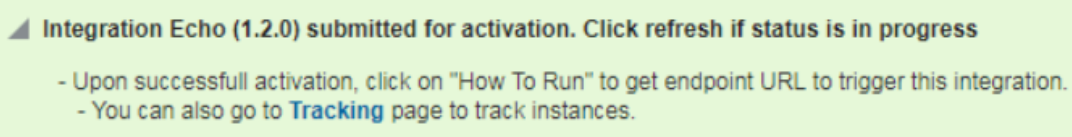
Element	Description
Contribute integration mappings to Oracle Recommendations	Click to enable the Oracle Recommendations Engine. Uses the collective intelligence to recommend which fields should be mapped while developing an integration. These recommendations are built based on the mappings contributed to Oracle Recommendations Engine anonymously. You can change this on the Recommendations page by selecting **Settings** > **Recommendations** in the navigation pane.
Enable Asserter Recording	Click to capture payloads and record instances for playing back and testing, See Test Integration Instances.
Enable tracing	Click to enable detailed tracing information in the activity stream. The tracing level has no impact on billing. When this checkbox is selected, detailed logging information about messages processed by this integration flow at runtime is collected. This can aid in troubleshooting issues. However, detailed tracing may also impact performance. To disable tracing, you must deactivate the integration, then reactivate it without selecting the **Enable tracing** checkbox. You can download the logs on the Dashboard page. If you want to enable payload tracing to log input and output payloads to the activity stream, click the **Include payload** checkbox. When this checkbox is selected, information from the payload is also written to the log files, which can be downloaded and viewed. For example, you see more detailed logging information about payload activity at several points in the integration flow, such as the payload prior to data mapping and the payload after data mapping. This action can present a security risk and also impact the performance of your system. This setting is not recommended for a production environment. See Activity Stream Logs Deleted. .

Element	Description
Activate & Deploy	Click to deploy a REST Adapter trigger-based connection to Oracle API Gateway. This option is only visible if you configured a connection with the Oracle API Gateway. See Connect to Oracle API Gateway.

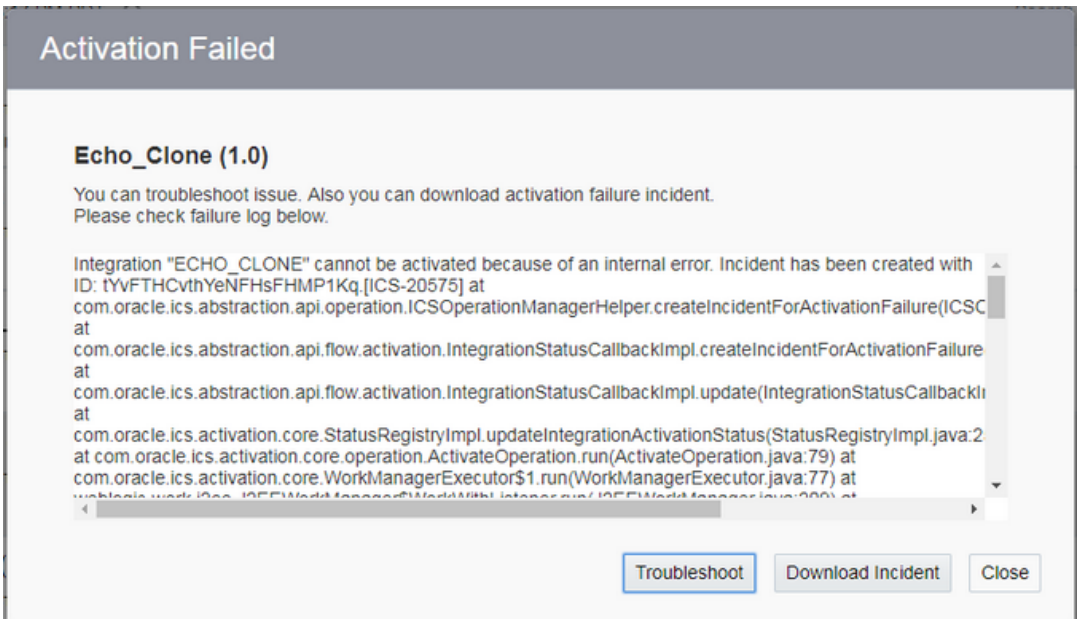
5. Select an activation option. The options available are based on the type of integration you are activating:
   a. Click **Activate** (if you are activating a nonscheduled integration).
   b. Click **Activate and Schedule** (if you are activating a scheduled integration).

      The Run *integration_name* page is displayed. Create a schedule for running this integration, and click **Save**. See Define the Integration Schedule.

A status message is displayed in the banner at the top of the page. For example:



6. If integration activation is unsuccessful, an **Activation Failed** warning icon is displayed in the banner. If you click the icon, a dialog is displayed with the option to download details about the activation failure incident.

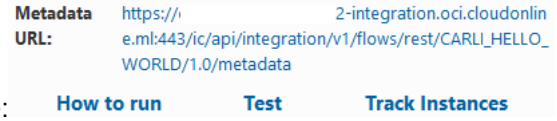


- If your integration includes a function that is not completely configured, an error message is displayed in the banner. You must complete configuration of this function before you can activate the integration. Click inside the integration and note the following errors/warnings:
  - An error icon is displayed on the function call action that uses the incomplete function. The **Error** panel on the right side of the integration canvas provides specific details about the incomplete function.

- A warning icon is displayed on the mapper that uses the inputs and outputs of this function. After completing function configuration, you must verify the input and output mappings before activating the integration.

If activation is successful, the status of the integration changes to **ACTIVE** in the row.

7. Click the 
icon to display details about how to run, test, and track instances for this integration.

For example: 

If you selected to enable tracing, the words **TRACE ENABLED** are displayed under the icon you clicked to activate the integration. If you click the integration instance on the Track Instances page, the  menu includes an option called **View Activity Stream** for viewing payload details.

To access the detailed trace logging information:

a. In the left navigation pane, click **Home** > **Monitoring** > **Integrations** > **Dashboards**.

b. Click **Download Diagnostic Logs** to download Oracle Integration logs and diagnostics logs.

8. View active integrations by clicking the integration name or selecting **View** from the  menu. The active integration is displayed with a message saying **View Only** in the banner at the top.

Note the following details about read-only mode:

- No **Save** button is displayed.
- There are no **Invokes**, **Actions**, or **Errors** icons.
- You can click through multiple parts of the integration to view configuration details, such viewing the business identifiers under the **Tracking** link, viewing the source-to-target and target-to-source mappings in the mapper, and viewing the configurations on the pages of the connection wizards, but you cannot modify anything.

Reactivate Integrations after a Connection Update

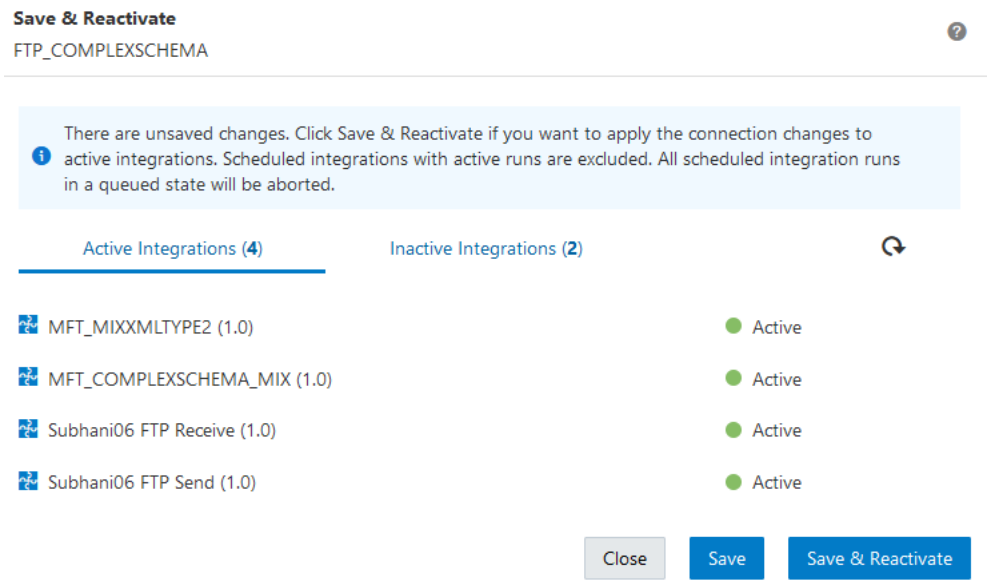
You can reactivate all active integrations together that use the same adapter connection after a connection update. This eliminates the need to manually deactivate and reactivate each integration individually to reflect the updated adapter connection. This functionality works for all adapters.

> **Note:**
>
> A maximum of 50 integrations can be reactivated after a connection update.

1. In the left navigation pane, click **Home** > **Integrations** > **Connections**.
2. Find and update the necessary connection. The number of integrations that use that connection is displayed in the banner.

3. Click **Test**, then click **Save**.

   The list of active and inactive integrations that use the updated connection are displayed on the right.



4. Click **Save & Reactivate** to apply the connection changes to all active integrations. Note the following:

   - You cannot select individual integrations to reactivate; all are reactivated.
   - Reactivation cancels all existing scheduled integrations, including the schedule's next run that's in a waiting state. The schedule must be restarted after deactivation to regenerate the canceled next run. Otherwise, the next run is skipped.
   - Any integrations using this connection that are currently listed as inactivate remain in that state.
   - If you click **Save**, your connection changes are saved, but no integrations are reactivated.
   - Deactivation of an integrations that subscribes to business events does not unsubscribe the business events.

   The page is refreshed to show integration activation progress (for example, the number of integrations that have been reactivated and the number of integrations remaining to be reactivated).

5. Click

   periodically to view progress.

   When complete, details about the reactivated integrations are displayed.

6. Click **Cancel** to close the page.

7. If any integration reactivations failed, go to the Integrations page to investigate.

General integration activation information is also provided. See Activate an Integration.

Deactivate an Integration

You can deactivate an integration in Oracle Integration to stop it from processing any new messages. If you want to modify an active integration, you need to deactivate it first. If the integration contains a business event subscription, a message is displayed asking if you want to delete the event subscription.

Deactivation is equivalent to undeployment of a project, which means all existing history, monitoring, and runtime data are lost. Oracle Integration does not support the notion of starting and stopping projects. With asynchronous patterns, the queue for the deactivated project is deleted and all messages associated with this queue are also deleted. Therefore, if there are pending requests unprocessed, they are lost after deactivation. The previous version is deactivated and all existing history, monitoring, and runtime data is lost.

1. In the left navigation pane, click **Home** > **Integrations** > **Integrations**.
2. Go to the row of the integration you want to deactivate.

   To view only active integrations, select **Active** in the
   
   filter. You can also filter by integration name or integration type (prebuilt, customized, or developed) to narrow down the list.
3. Click the
   
   icon to deactivate the integration.
4. Click **Deactivate** in the dialog that appears. A deactivation progress bar is displayed at the top of the dialog.

> **Note:**
>
> If you attempt to deactivate a scheduled integration, any active or paused schedule for this integration is also stopped as part of deactivation.

Integration deactivation can take time to complete after you initiate the request. During this time period, there may still be in-flight or queued requests being processed. These processing instances fail as the corresponding integration artifacts and resources are removed and undeployed.

You can observe different processing errors for the failed instances depending on the point of execution during the deactivation. You can also observe the failed instances occurring after the deactivation request.

Deactivate an Integration with Business Events

If the integration to deactivate contains a business event subscription, a message is displayed asking if you want to delete the event subscription. If you select to delete the event subscription, the integration does not receive any events after it is reactivated.

If you do not want to delete the event subscription, the events in this integration are resent if the integration is activated within six hours.

Enable or Disable Tracing on Active Integrations

You can enable or disable tracing on activated integrations without re-activating them. You can perform these actions on individual integrations or globally on all integrations.

> **Caution:**
>
> Enabling tracing at the integration or global level impacts performance. Oracle strongly suggests that you don't enable this option in production environments. Including the payload in tracing impacts the performance of your system and may violate your company's data retention rules by storing sensitive information in the payload trace.

1. In the navigation pane, click **Home** > **Settings** > **Integrations** > **Tracing**.
2. Select an option below to configure the tracing level. The tracing level has no impact on billing.

Element	Description
Integration Level	Select to enable or disable tracing at the individual integration level on the Integrations page. This enables you to select **Tracing** from the  menu of an individual integration on the Integrations page and choose to disable tracing or enable tracing and include the payload in the integration. If you enable tracing, details are displayed in the activity stream.
Global Tracing On	Select to globally turn on tracing for all activated integrations. After selecting and saving this option, you cannot enable or disable tracing at the individual integration level on the Integrations page. The following message is displayed for all integrations on the Integrations page. `TRACE WITH PAYLOAD` Tracing details are displayed in the activity stream. If you attempt to disable tracing for an individual integration on the Integrations page by selecting **Tracing** from the  menu, a dialog is displayed indicating that global tracing is currently enabled for all active integrations. You must first return to this page and select **Integration Level** to enable or disable tracing at the integration level.

Element	Description
Global Tracing Off	Select to globally turn off tracing for all activated integrations. After selecting and saving this option, you cannot enable or disable tracing at the individual integration level on the Integrations page. If you attempt to enable tracing for an individual integration on the Integrations page by selecting **Tracing** from the  menu, a dialog is displayed indicating that global tracing is currently disabled for all active integrations. You must first return to this page and select **Integration Level** to enable or disable tracing at the integration level.

Tracing can impact the activity stream logs. See Activity Stream Logs Deleted.

Test REST Adapter Trigger Connection-Based Integrations

You can test app driven orchestration integrations designed with a REST Adapter trigger connection from the Test page in Oracle Integration. This eliminates the need to use third-party software to run this type of integration.

1. In the left navigation pane, click **Home** > **Integrations** > **Integrations**.
2. Go to the row of the REST Adapter trigger connection-based integration to run.
3. Click  , then click **Test** on the message that is displayed.

   The Test Integration page is displayed with the following sections:

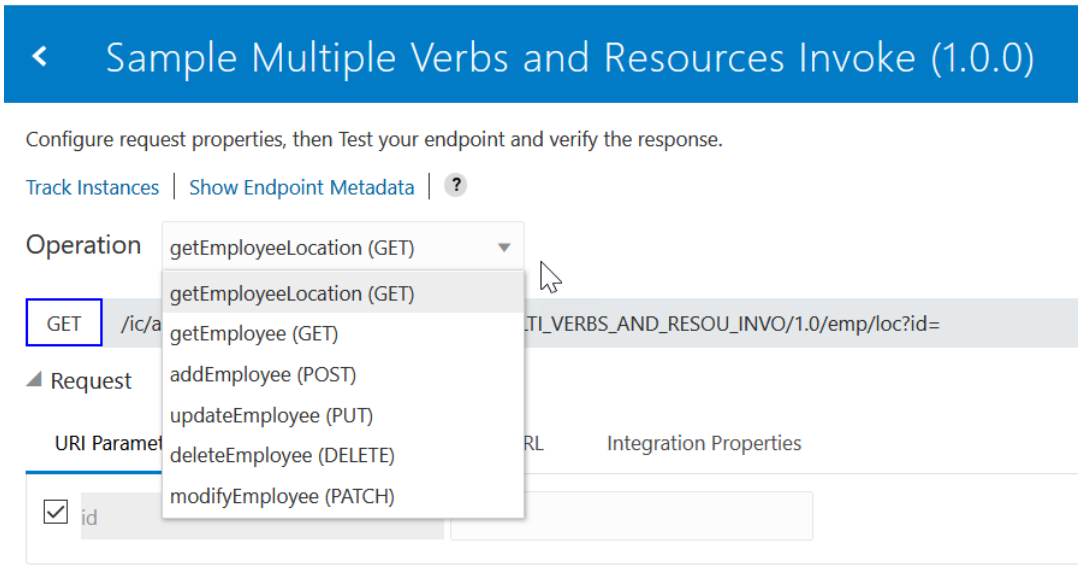
   - Operation Section
   - Request and Response Sections

   The **Operation** and **Request** sections have the endpoint's metadata populated.

Operation Section

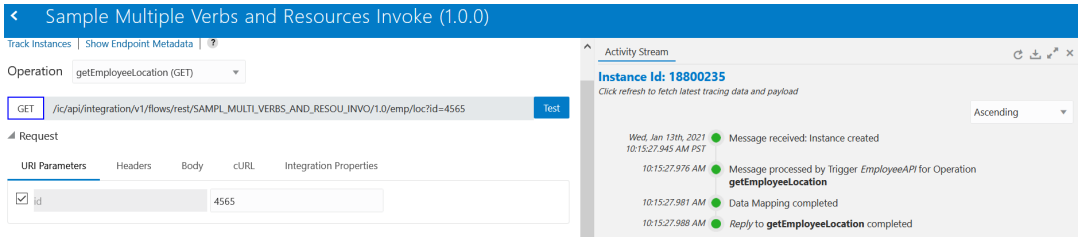
If the REST Adapter trigger connection-based integration is configured with multiple operations, the **Operation** section is displayed. This section contains the **Operation** list for selecting the appropriate operation, along with the HTTP method and relative URI (for the selected operation in case of multiple operations). If your integration contains only a single operation, this list is not displayed.

1. Select the available operation to test.



2. Enter a value and click **Test** to invoke the integration (for this example, a value is specified for **id**).

   The activity stream results are displayed.



Request and Response Sections

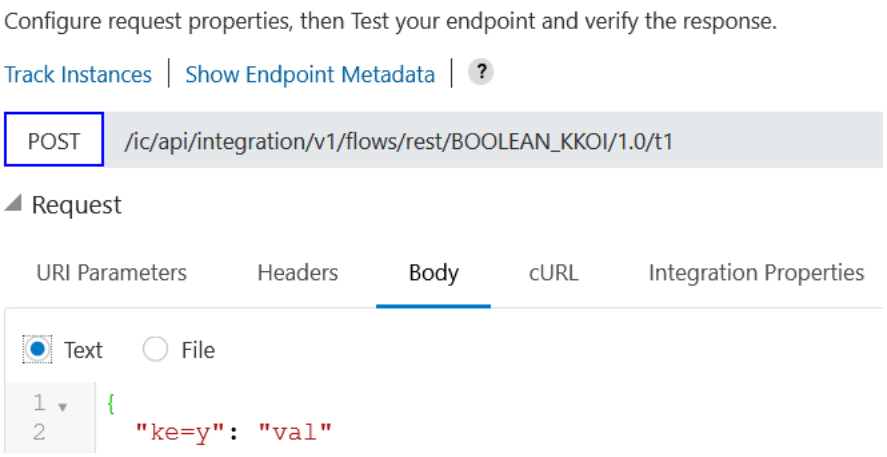
The **Request** and **Response** sections enable you to trigger and view response output for your REST Adapter trigger connection-based integrations. Links are provided at the top to access the Track Instances page and the endpoint metadata.

The **Request** section consists of the following fields:

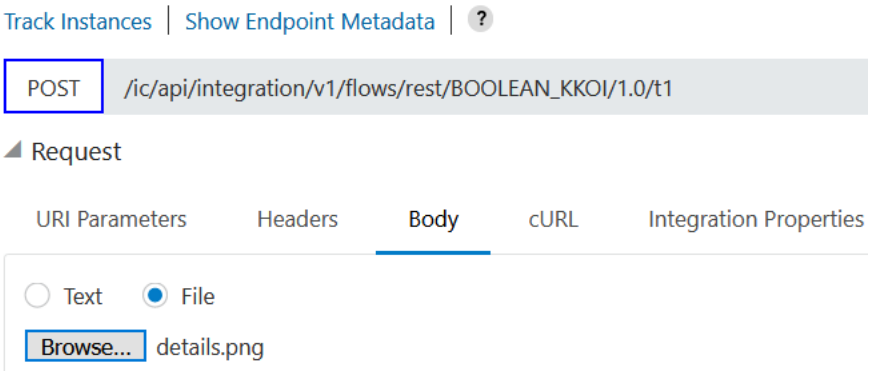
- **URI Parameters**: Shows the list of expected path (or template) and query parameters. You can enter data in this field.
- **Headers**: Shows the custom headers, including **Accept** and **Content-Type** based on the integration configuration.
- **Body**: Enables you to view the message body contents or upload a file. Support is provided for uploading payloads with multipart payloads or for uploading nontextual payloads such as PDF or image files.
  - If the request operation does not have a body, the following message is displayed.

    ```
    This request does not have a body
    ```

  - If the request operation includes a body, its contents are automatically displayed. For example:



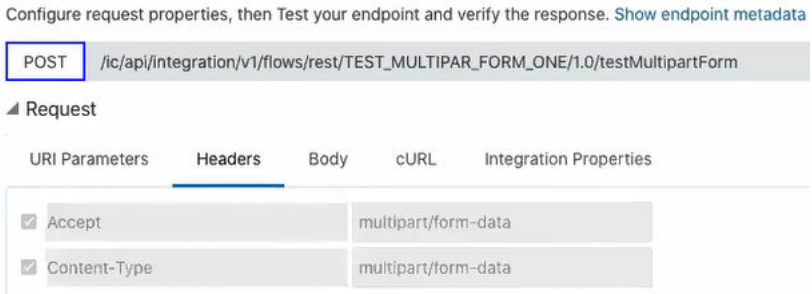
– If you want to browse for and upload a binary or nontextual payload (such as a PDF or image file), click **File**. For this example, an image file is uploaded.



– For endpoints with multipart payloads, a form is displayed under the **Body** field that can be used to provide a value for each part in the multipart body. Two multipart attachments types are supported in the request payload:

  * multipart/form-data
  * multipart/payload with structured data such as JSON and XML:
    * multipart/form-data
    * multipart/mixed

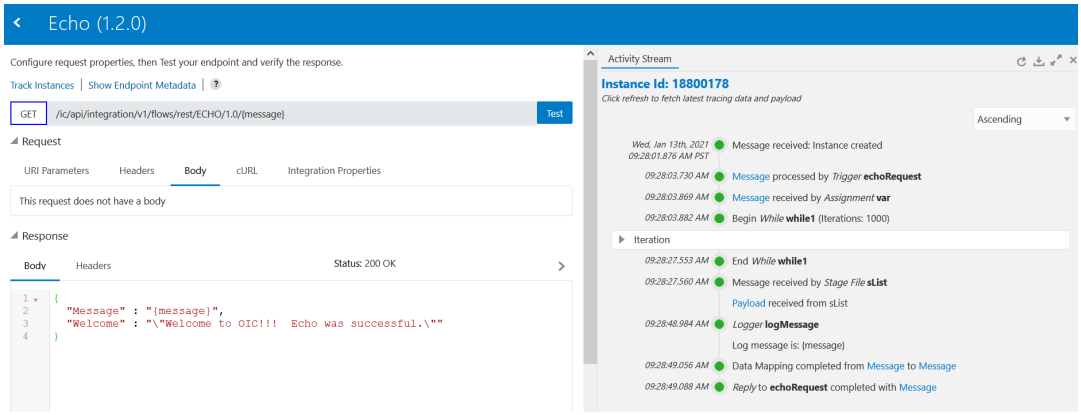
For the following example, two headers for multipart/form-data are visible under the **Headers** tab.



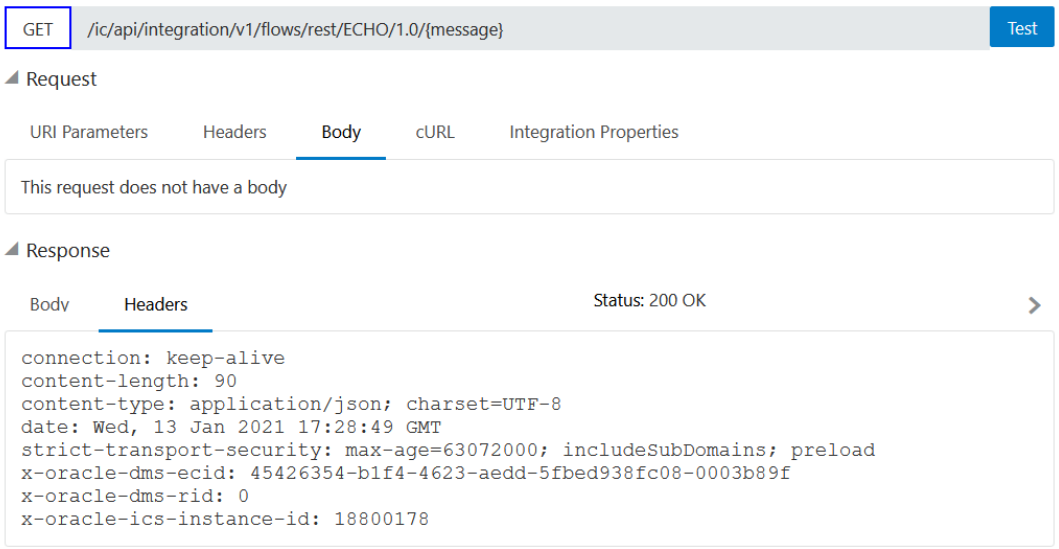
Use cases for both multipart attachment types are provided. See Test REST Adapter Trigger-Based Integrations with Multipart Attachments on the Test Page.

- **cURL**: Enables you to copy the equivalent `curl` command. The `curl` command is generated based on the endpoint's metadata and your provided input.

1. Click **Test** to invoke the integration.
The activity stream for this instance is displayed on the right side of the page. You can also access the instance in the Track Instances page by clicking **Track Instances** above the operation.



2. Check the **Response** section for response details.
The **Response** section consists of the response body (if any), headers included in the body, HTTP status of the headers (`200 OK` for this example), and instance ID of the triggered integration (if any).



3. Scroll to the right of **Status** and note the instance ID link. This link takes you to the details page for the instance under the Track Instances page.
4. Click the **Body** tab in the **Response** section and place your cursor in the response message.

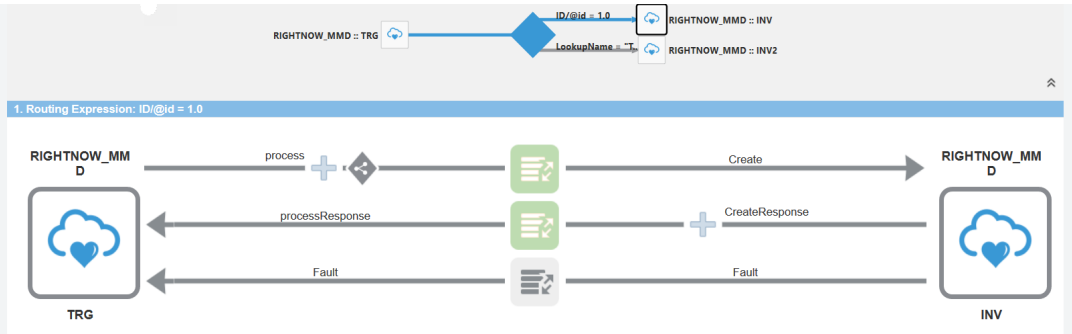
5. From the menu, click either **Download** to download the response payload in a file or **Copy to Clipboard** to copy the response payload to your clipboard.

Convert a Basic Routing Integration to an App Driven Orchestration Integration

You must convert basic routing integrations to app driven orchestration integrations. Basic routing integrations offer limited action capabilities and are being deprecated. App driven orchestration integrations provide greater flexibility with more action capabilities.

> **Note:**
>
> Basic routing scheduled integrations, publish to OIC integrations, subscribe to OIC integrations, and asynchronous integrations with a delayed response cannot currently be converted.

1. In the left navigation pane, click **Home** > **Integrations** > **Integrations**.
2. Click to filter integrations by style.
3. Go to the row of the basic routing integration to convert. For this example, the following basic routing integration is selected for conversion. Note that routing paths for two different ServiceNow Adapter invoke endpoints have been defined.



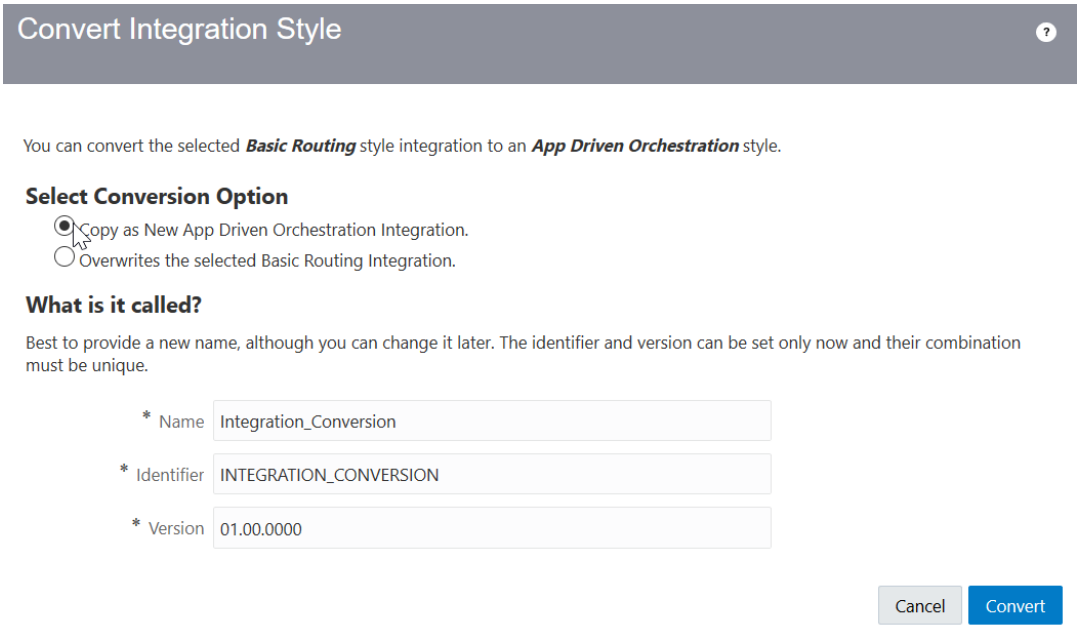
4. Select **Convert** from the menu.

> **Note:**
>
> The option is not available for basic routing integrations that are active or locked.

5. In the **Select Conversion Option** section, select the type of conversion to perform.
   - **Copy as New App Driven Orchestration Integration**: Creates a new app driven orchestration integration copy from the basic routing integration. This enables both integration styles to exist together until you are ready to delete the basic routing integration style.

- **Overwrites the selected Basic Routing Integration**: Overwrites the existing basic routing integration with a new app driven orchestrated integration.

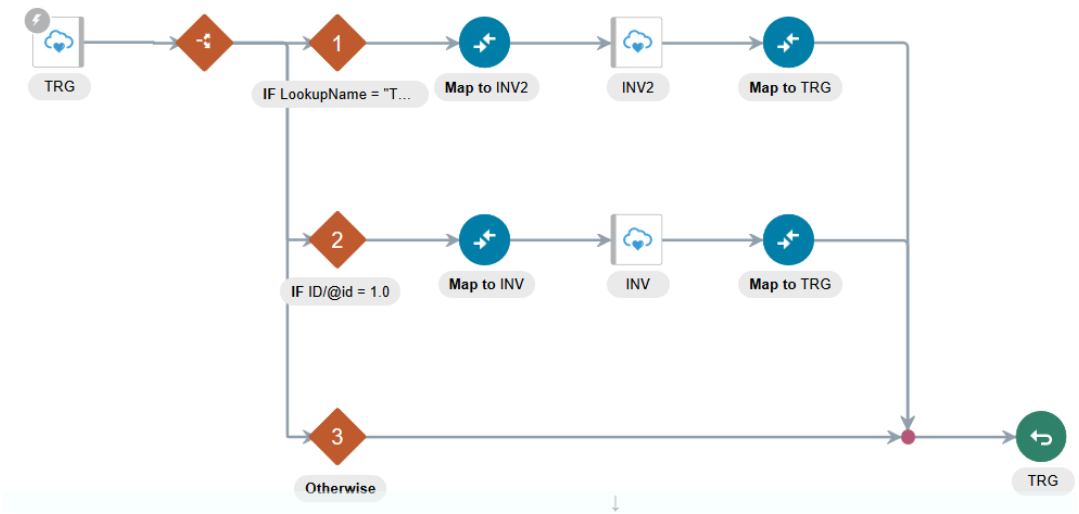
6. Optionally enter a new name for the integration that can always be changed later.
7. Accept the default values or enter new values for the identifier and version. These cannot be changed later. Their combination must be unique.



8. Click **Convert**.
When conversion completes, the following message is displayed:

```
CONFIRMATION
Integration Integration_Name (version_number) was converted successfully.
```

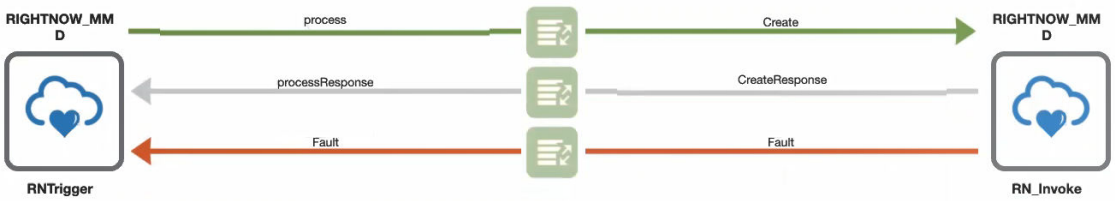
9. Open the converted integration. Note that the routing path in the basic routing integration was converted to a switch action that includes the two different ServiceNow Adapter invoke endpoints.



10. Activate the integration. See Activate an Integration.

Faulted Integrations Behavior After Conversion

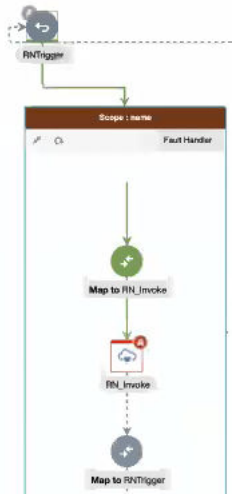
There is a change in behavior for faulted integrations after basic routing to app driven orchestration integration conversion is complete. When a synchronous basic routing integration fails, a failed tracking instance is created. The failed integration is visible on the Tracking Details page for that integration under **Home** > **Monitoring** > **Integrations**.



If you select **View Errors** from the menu, the following error is displayed.

```
Error while invoking target service "integration_name" for sync flow.
The Error was raised back to the calling client.
```

After conversion to an app driven orchestration integration, the tracking instance is marked as succeeded for this integration. This is because the failure is handled in a scope fault handler.



If you select **View Activity Stream** from the menu, the following message is displayed.

```
Error Message is - Error has been recovered. No message to display.
```

Therefore, the integration is marked as succeeded instead of failed.

Edit and Replace Dependent Resources

You can edit and replace dependent resources in a single integration or all integrations in a package.

Topics:

Edit and Replace Dependent Resources in an Integration

The Configuration Editor page lists all the dependent resources used by all integrations in a specific package or any standalone integrations that are not part of a package.

Edit and Replace Dependent Resources

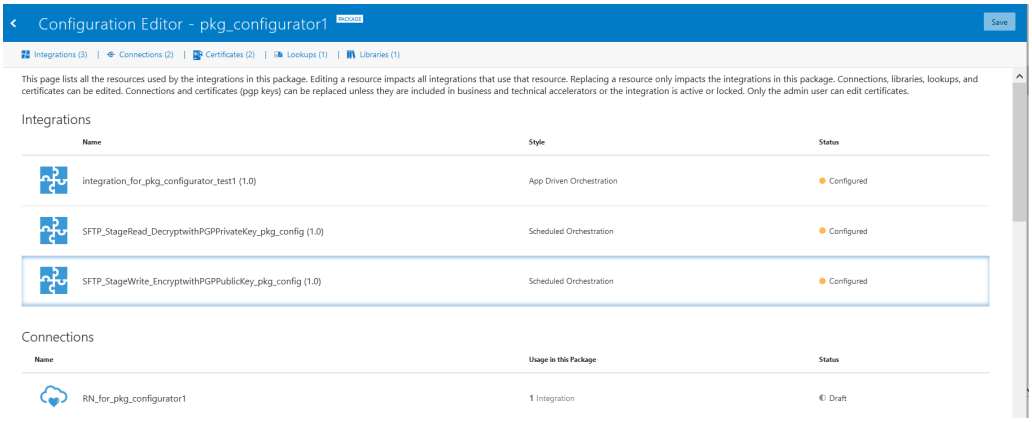
Dependent resources consist of integrations, connections, libraries, lookups, and PGP keys. You can edit integrations, connections, libraries, lookups, and PGP keys. You can replace connections and PGP keys unless they are included in business and technical accelerators or the integration is active or locked. Only the admin user can edit PGP keys. You can also update integration property values for integrations in which properties have been defined and add schedules to scheduled integrations.

Editing a dependent resource impacts all integrations that use that resource. Replacing a dependent resource only impacts the integrations in the specific package or the standalone integration that is not part of any package.

1. In the left navigation pane, click **Home** > **Integrations** > **Integrations**.
2. Select the integration in which to edit or replace resources in either of two ways:
   a. Go to the row of the integration.
   b. Select **Configure** from the  menu.

   Or:

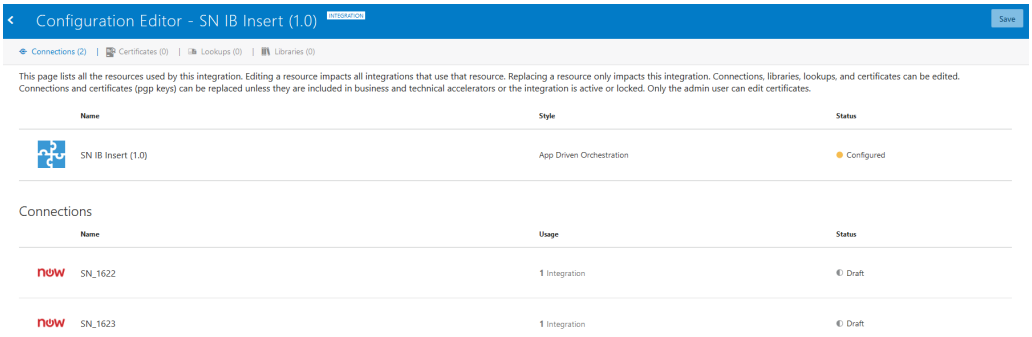
   a. Select **Import**.
   b. Browse for and select the integration to import, then select **Import and Configure**.
   - If the integration you selected is part of a package, the Configuration Editor page is displayed with the package name in the banner and links to the integrations, connections, certificates (PGP keys), lookups, and libraries included in the package. The integration you selected is highlighted on the page.



A message automatically slides out from the upper right corner to identify the integration name and its associated package.

Integration_name (*version_number*) belongs to the package *package_name*. It is highlighted
below and will be configured as part of this package.

- If the integration you selected is not part of a package, the Configuration Editor page is displayed with links at the top to the connections, certificates (PGP keys), lookups, and libraries included in the package.

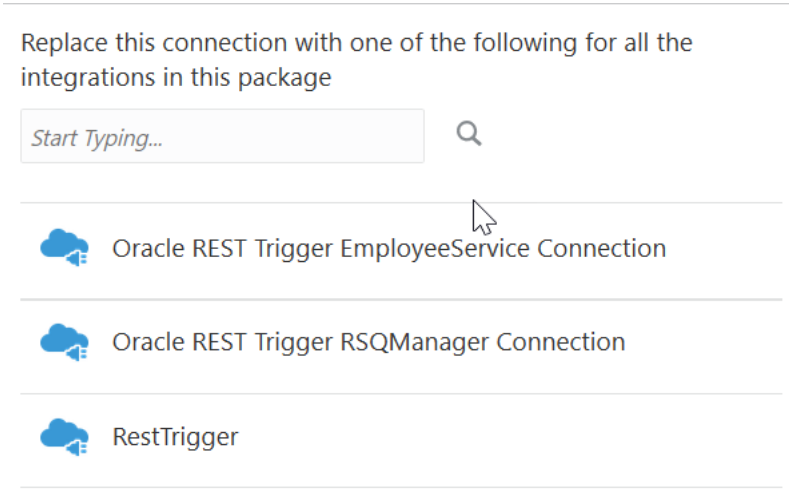


3. Click the link at the top for the type of resource you want to edit or replace.
4. Click

at the far right of the row to access the corresponding edit page of the resource (for example, the Connections page, Lookup page, JavaScript Library page, or Update Certificate page). Editing a resource impacts all integrations that use that resource.
   a. Make the necessary edits and save your changes.
      i. If editing an integration, click **Save**, then click **Close** to return to the Configuration Editor page.
      ii. If editing a connection, lookup, library, or certificate, click **Save**, then click

to return to the Configuration Editor page.

5. Click  at the far right of the row to replace a connection or PGP key. For example, to replace a connection, the following page is displayed. Only connections of the same role type (trigger, invoke, or trigger and invoke) are displayed for selection.



- A connection resource can only be replaced by another resource of the same role (trigger, invoke, or trigger and invoke).
- The connection status must be **Configured**. You cannot replace a connection with a connection whose status is **Draft**. If a compatible connection resource does not exist, a message is displayed.
- A PGP key must be the same type of key. For example, a public PGP key can only be replaced with another public PGP key, and not a private PGP key.

If you replace a resource, the information is persisted in the details section under the header. For example:

```
The connection for the integrations in this package was changed from
connection_name. Revert
```

You can revert the replaced resource to the original by clicking the **Revert** link in the message.

6. Click  at the far right of the row to display the following options.

Option	Description
Add Schedule	If this is a scheduled integration and a schedule has not been created, this option is displayed. Select to open the Schedule page. You can add a schedule to the integration. See Define the Integration Schedule.

Option	Description
Schedule	If this is a scheduled integration and a schedule has been created, select to open the Schedule and Future Runs page. You can edit the schedule for the integration. See Edit an Integration Schedule.
Update Property Values	Select to slide open the Update Property Values dialog on the right side of the page. A message is displayed if properties have not yet been defined. You can configure integration properties outside of the regular integration design. At runtime, these integration properties take effect and override the values you configured at design time. See Override Design-Time Properties in an Integration.

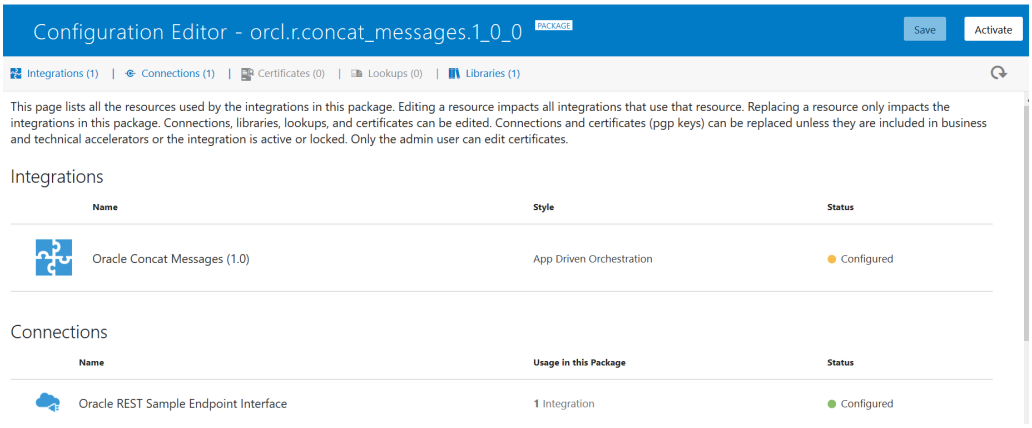
7. Click a resource in the **Style** column to see which integrations are using the resource.
8. When you have completed editing resources on the Configuration Editor page, click **Save**.

Activate or Deactivate an Integration

If your integration type is a recipe or an accelerator, you can activate or deactivate the integration from the Configuration Editor page.

1. In the left navigation pane, click **Home** > **Integrations** > **Integrations**.
2. Go to the row of the integration to active or deactivate. The integration type must a recipe or accelerator. Use  to filter the search for integration types.
3. Select **Configure** from the
   
   menu.
   The current status of integration activation is displayed in the upper right corner:
   - If **Activate** is displayed, the integration is not activated.
   - If **Deactivate** is displayed, the integration is activated.
4. If you want to activate the integration, perform the following steps.
   a. Click **Activate** in the upper right corner to access a page for activating the integration.

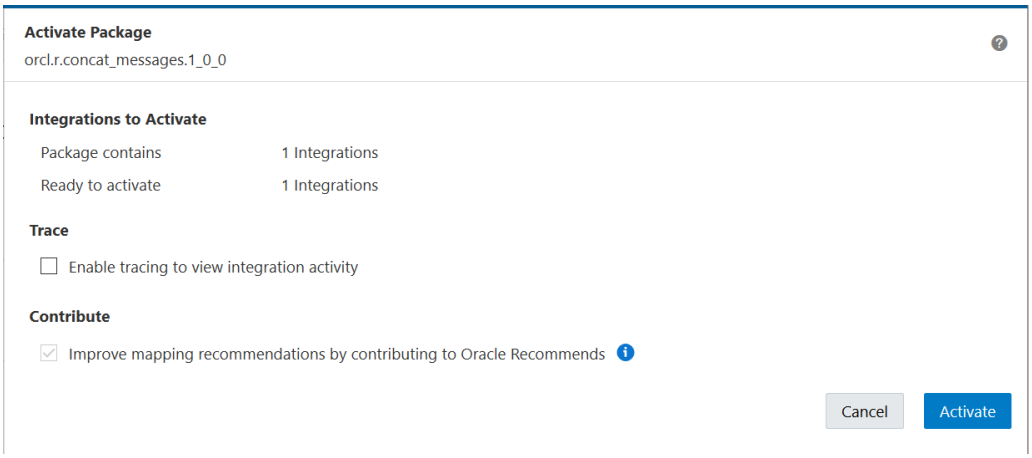
> **Note:**
>
> You can also click
> 
> at the end of the row. If the integration is not completely configured for activation, a message describing what tasks must be performed is displayed.



If the selected integration is part of a package, status is provided for all other integrations in the package, and not just the integration you selected.

If the integration cannot be activated, the **Not ready to activate** field is displayed. Expand

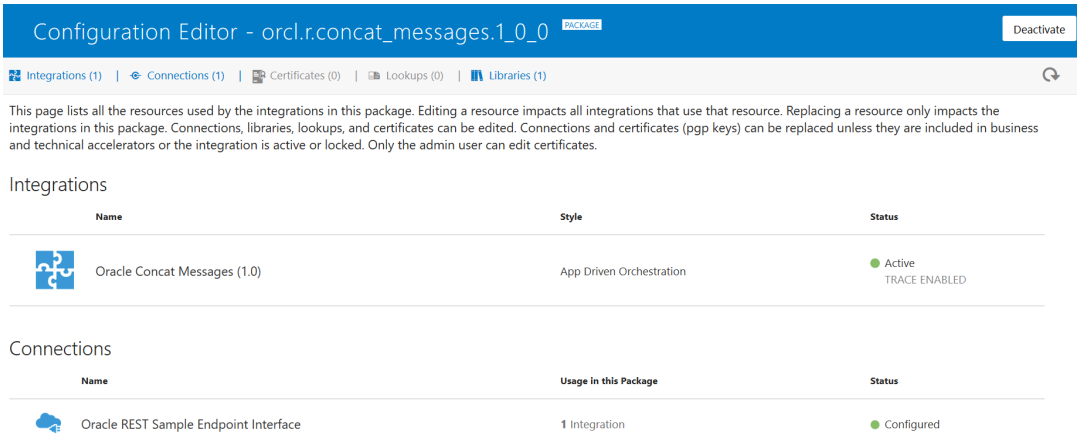
to view details about what must be resolved before this integration can be activated.



b. Provide responses for the tracing and mapping recommendations fields on the page.

c. Click **Activate**.

5. If you want to deactivate the integration, click **Deactivate**.



Edit and Replace Dependent Resources in a Package

The Configuration Editor page lists all the dependent resources used by all integrations in a specific package.

- Edit and Replace Dependent Resources
- Activate and Deactivate All Integrations in a Recipe or Accelerator Package

Edit and Replace Dependent Resources

Dependent resources consist of integrations, connections, libraries, lookups, and PGP keys. You can edit integrations, connections, libraries, lookups, and PGP keys. You can replace connections and PGP keys unless they are included in business and technical accelerators or the integration is active or locked. Only the admin user can edit PGP keys.

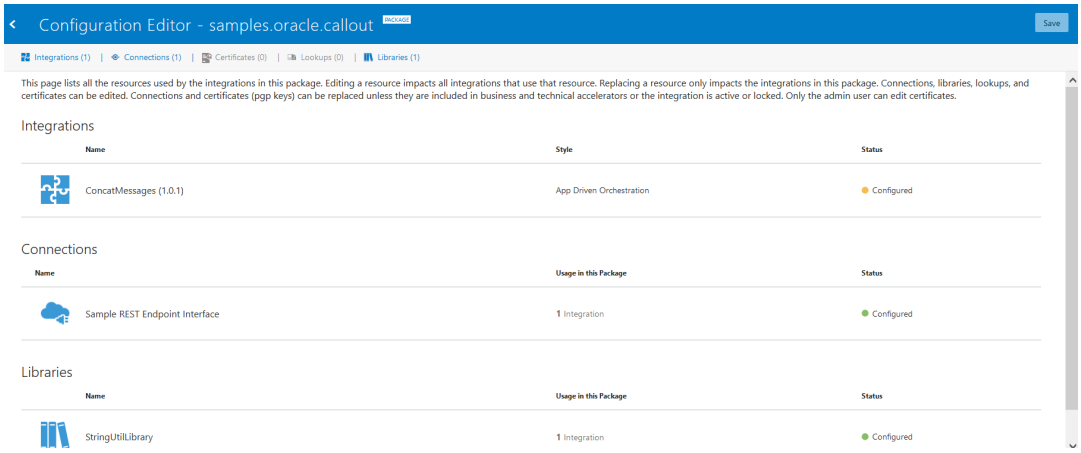
Editing a dependent resource impacts all integrations that use that resource. Replacing a dependent resource only impacts the integrations in the specific package.

1. In the navigation pane, click **Home** > **Integrations** > **Packages**.
2. Select the package in which to edit or replace resources in either of two ways:
   a. In the banner, click **Import**.
   b. Browse for and select the PAR file when prompted.
   c. Click **Import and Configure**.

   Or:

   a. Go to the row of the package.
   b. Click
   .

   The Configuration Editor page is displayed with the package name in the banner and links to the integrations, connections, certificates (PGP keys), lookups, and libraries included in the package.



3. Click

at the far right of the row to access the corresponding edit page of the resource (for example, the integration canvas, Connections page, Lookup page, JavaScript Library page, or Update Certificate page).

   a. Make the necessary edits and save your changes.

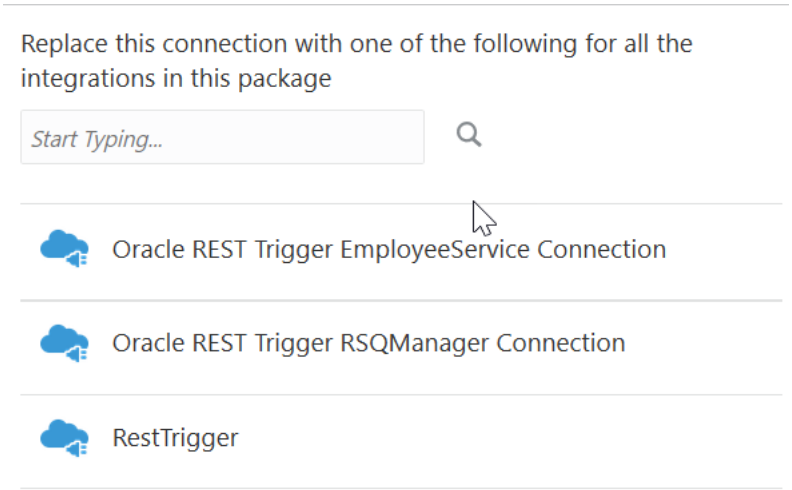
      i. If editing an integration, click **Save**, then click **Close** to return to the Configuration Editor page.

      ii. If editing a connection, lookup, library, or certificate, click **Save**, then click

to return to the Configuration Editor page.

4. Click

at the far right of the row to replace a connection or a PGP key across all the integrations in a package. For example, to replace a connection, the following page is displayed. Only connections of the same role type (trigger, invoke, or trigger and invoke) are displayed for selection.



- You can replace connections used by inactive integrations in a package. For example, if a package contains five integrations, and one integration is inactive, you can replace the connection for only that integration. The connections in the other four integrations cannot be replaced.
- You cannot replace connections used by active, currently being activated, currently being deactivated, and locked integrations in a package. For example, if a package contains five integrations, and two integrations are active, one is currently being activated, and two are inactive, you can replace the connections for only the two inactive integrations. The connections in the other three integrations cannot be replaced.
- A connection can only be replaced by another connection of the same role (trigger, invoke, or trigger and invoke).
- The connection status must be **Configured**. You cannot replace a connection with a connection whose status is **Draft**. If a compatible connection resource does not exist, a message is displayed.
- A PGP key must be the same type of key. For example, a public PGP key can only be replaced with another public PGP key, and not a private PGP key.

If you replace a resource, the information is persisted in the details section under the header. For example:

```
The connection for the integrations in this package was changed from
connection_name. Revert
```

You can revert the replaced resource to the original by clicking the **Revert** link in the message.

5. Click

at the far right of the row to display the following options.

Option	Description
Add Schedule	If this is a scheduled integration and a schedule has not been created, this option is displayed. Select to open the Schedule page. You can add a schedule to the integration. See Define the Integration Schedule.
Schedule	If this is a scheduled integration and a schedule has been created, select to open the Schedule and Future Runs page. You can edit the schedule for the integration. See Edit an Integration Schedule.
Update Property Values	Select to slide open the Update Property Values dialog on the right side of the page. A message is displayed if properties have not yet been defined. You can configure integration properties outside of the regular integration design. At runtime, these integration properties take effect and override the values you configured at design time. See Override Design-Time Properties in an Integration.

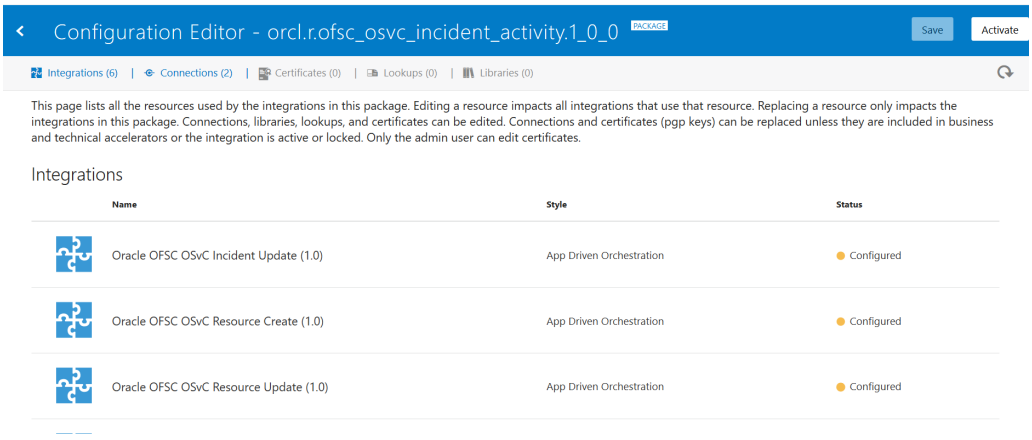
6. Click a resource in the **Style** column to see which integrations are using the resource.

Activate and Deactivate All Integrations in a Recipe or Accelerator Package

If your package type is a recipe or an accelerator, you can activate and deactivate all integrations in the package together. You *cannot* activate and deactivate user-created packages (known as developed packages). See Integration Accelerators and Recipes in *Getting Started with Oracle Integration Generation 2*.

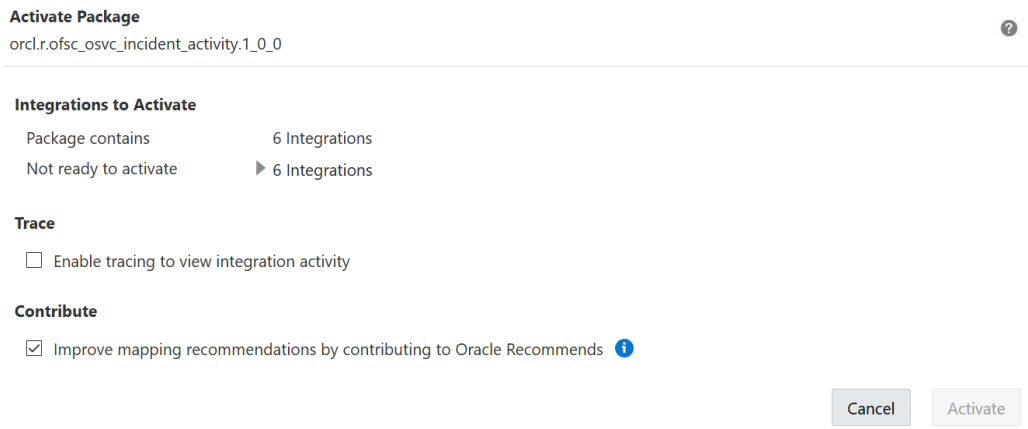
1. Go to the row of the recipe or accelerator package. Use to filter the search for package types.
2. Click
   .
   The current status of package activation is displayed in the upper right corner:
   - If **Activate** is displayed, the package is not activated.
   - If **Deactivate** is displayed, the package is activated.
3. If you want to activate the package, perform the following steps.
   a. Click **Activate** in the upper right corner to access a page for activating all integrations in the package together.

> **Note:**
>
> If you only want to activate a single integration in the package, you can instead click
>
> at the end of the row. If the integration is not completely configured for activation, a message describing what tasks must be performed is displayed.



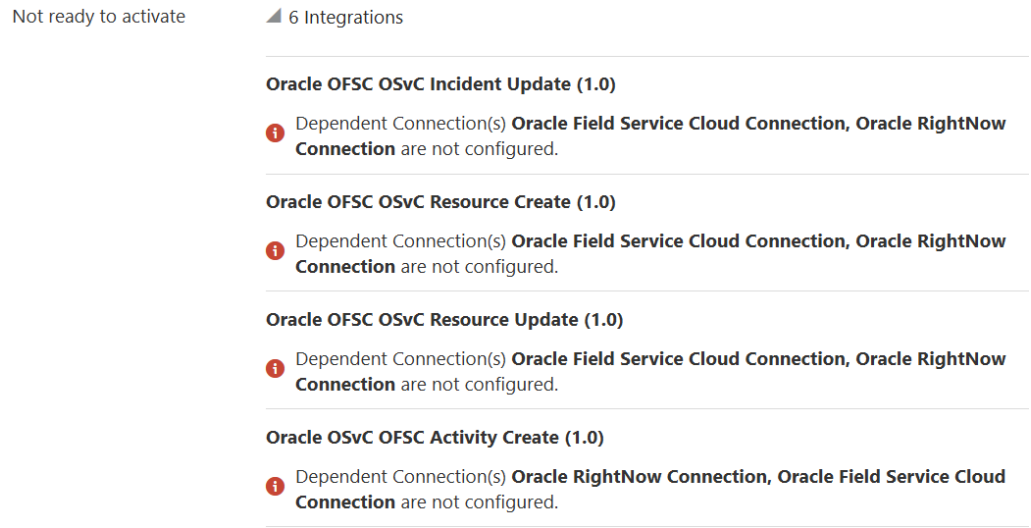
Details about the activation status of all integrations in the package are displayed:

- If at least one integration in the package is ready to activate, the **Activate** button is enabled. Any integrations that cannot be activated are displayed under the **Not ready to activate** button.
- If no integrations are ready to activate (as is the case with the following example), the **Activate** button is disabled. Details are displayed under the **Not ready to activate** button.



b. Expand

to view details about the integrations. For this example, adapter connections require configuration before these integrations can be activated.



c. Resolve any issues.

d. Provide responses for the tracing and mapping recommendations fields on the page.

e. Click **Activate**.
Depending upon the number of integrations in the package, activation can take several minutes.

4. If you want to deactivate the package, click **Deactivate**.
Depending upon the number of integrations in the package, deactivation can take several minutes.

Manage Packages

You can group integrations into a package. When you import or export the package to or from Oracle Integration, all integrations in that package are imported or exported.

Topics:

When you create an integration, you can also create a package or select an existing package in which to include the integration. See Create Integrations.

View the Integrations in a Package

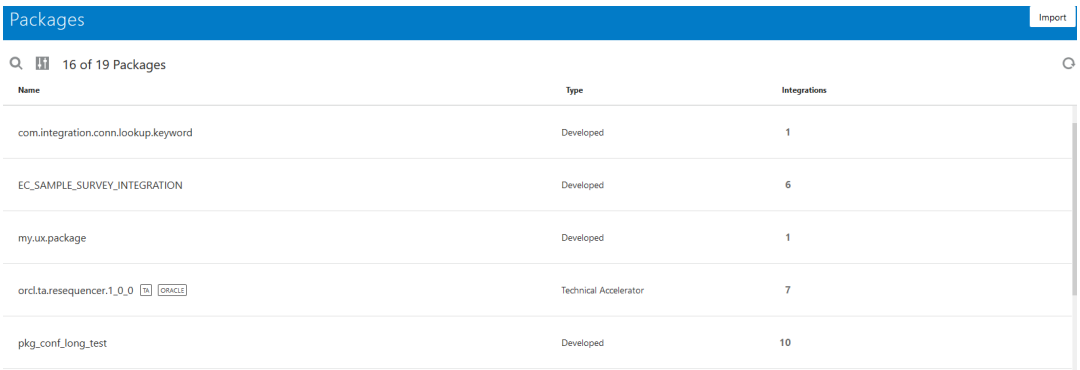
You can view the integrations included in a package.

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

   The Packages page is displayed. The package names and the number of integrations included in each package are displayed. You can filter the display of packages with the

   icon. If you have not yet created or imported a package into Oracle Integration, this page is empty.



2. Click the name of the package or click the

   menu.

   The integrations included in that package and their current states are displayed (for example, pending activation or active).

3. Click **Close**.

4. In the navigator pane, click **Integrations** to access the Integrations page for viewing these integrations.

Import a Package

You can import a package of integrations into Oracle Integration from the Packages page. The Packages page enables you to import packages that you or other users have created. To import packages that consist of integrations that are prebuilt by Oracle, you must go to Oracle Marketplace.

> **Note:**
>
> You cannot import a package if it contains activated integrations. Ensure that all integrations are deactivated before including them in a packages archive (PAR) file.

1. In the navigation pane, click **Home** > **Integrations** > **Packages**.
2. In the banner, click **Import**.
3. Browse for and select the PAR file when prompted.
4. Click **Import**.

   The package is added to the list on the Packages page.

You can import a prebuilt packages from Oracle Marketplace. See Import a Prebuilt Integration.

Export a Package

You can export a package of integrations from Oracle Integration on the Packages page. This action exports all the integrations included in that package.

1. In the navigation pane, click **Home** > **Integrations** > **Packages**.
2. Go to the row of the package to export.
3. Select **Export** from the
   
   menu.
4. Click **Export** when prompted to confirm your selection.
5. Save the package (PAR) file of integrations to a file system location when prompted. The individual integrations inside the PAR file are exported as integration archive (IAR) files.

Update a Package

You can update the package in which your integration is included. For example, you can create a new package for your integration or move your integration to an existing package.

1. On the Integrations page, find the integration of the package that you want to update. The integration must not be active.
2. Go to the row of the integration to edit.
3. Select **Edit** from the
   
   menu.

   The integration is displayed.

4. From the  menu, select **Primary Info**.
5. In the **Package** field, enter a new package name or enter an existing package name (as you type the initial letters, the existing package is displayed) to move your integration to an existing package.
6. Click **Save**, then click **Close**.
7. In the navigation pane, click **Packages**.
8. Click the package name you specified to see your integration.

Delete a Package

You can delete a package. This action deletes the package and all integrations included in that package.

1. In the navigation pane, click **Home** > **Integrations** > **Packages**.
2. Go to the row of the package to delete.
3. Select **Delete** from the
   
   menu.
4. Click **Delete** when prompted to confirm your selection. The package and all of its integrations are deleted.

If you want to delete a package, but not the integrations, navigate to each integration listed in the package and clear the **Package** field or replace the package name with a different name. When the last integration is removed from the package, the package is automatically deleted. See Update a Package.

Import and Export Integrations

You can import and export integrations to share them between Oracle Integration environments.

See the following topics:

- Import an Integration
- Import a Prebuilt Integration
- Export an Integration

Import an Integration

You can import integrations that were previously exported as a JAR file from Oracle Integration.

> **Note:**
>
> - You cannot import Oracle Integration Classic (user-managed) and Oracle Integration integrations into Oracle Integration Cloud Service.
> - You cannot manually edit an exported integration (IAR) file outside of Oracle Integration and then import it into another instance. For example, if you manually update a connection name in an exported integration, then import it, the connection created cannot be updated or deleted. If you need to modify the file, contact Oracle Support Services.

To import an integration:

1. In the left navigation pane, click **Home** > **Integrations** > **Integrations**.
2. In the banner, click **Import**.
3. In the Import Integration File dialog, click **Browse** to navigate to and select the file to import.
4. If the integration includes Oracle Asserter recordings that you want to import, click **Include Asserter Recordings (if any)**.
5. Click **Import**.

   If an integration already exists with the same identifier and version, you must confirm whether to overwrite the existing integration.

The imported integration appears in the Integrations list and you can customize or activate it.

> **Note:**
>
> Even though the **Activate** icon is enabled after you import an integration, you must first configure your connection endpoints. If you do not, you receive an error when trying to activate the integration. See Edit a Connection.

Import a Prebuilt Integration

You can import prebuilt integrations into your Oracle Integration environment.

There are two types of prebuilt integrations:

- User-created integrations. These are integrations that you or another user created. See Import an Integration.
- Oracle-created integrations from the Oracle Integration Home page or from the Oracle Cloud Marketplace as part of a package. Integrations from the Home page are designated with a **RECIPE** message that is displayed next to the integration name on the Integrations page. See Get Familiar with the Home Page in *Getting Started with Oracle Integration Generation 2* and the Oracle Cloud Marketplace.

Export an Integration

Once you create an integration, you can export that integration as a JAR file for use in other Oracle Integration environments or import the integration into Oracle JDeveloper to perform an advanced XSLT mapper task (for example, creating variables or using templates) that you cannot perform in the mapper. After mapper editing in Oracle JDeveloper is complete, the mapper file can then be imported back into Oracle Integration. You can export an integration from either the Integration Designer or from the Integrations list. You can also export a locked integration.

> **Note:**
>
> You cannot import Oracle Integration Classic (user-managed) and Oracle Integration integrations into Oracle Integration.

To export an integration:

1. In the left navigation pane, click **Home** > **Integrations** > **Integrations**.
2. Go to the row of the integration to export.
3. Select **Export** from the
   
   menu.
4. In the dialog that appears, select **Save File**, and then click **OK**.
5. Save the file to the location you want.

The file is saved with a name that consists of the identifier plus the version number, and an IAR extension.
You can import the exported integration into the XSL Map Editor in Oracle Service Bus. See Import a Map File into Oracle JDeveloper.

> **Note:**
>
> Lookups referenced using the `lookupValue` function in the Expression Builder are included in the exported integration JAR file. When you import the integration, the referenced lookups are also imported and are visible in the Expression Builder.

Regenerate a WSDL File for Integrations

After you clone an integration, customize a prebuilt integration, or import an existing integration into Oracle Integration, you update the connection information (WSDL, username, and password) according to the requirements of your integration environment. If the connection WSDL you specify contains any custom fields or if the connection WSDL is updated with a different version, they are not displayed in the mapper. To get custom fields or updated fields to appear in the mapper, you must regenerate the endpoint in Oracle Integration.

As an example, you may have a scenario in which the WSDLs with one of your connections (for example, a Salesforce connection) change frequently and you must be able to uptake the latest WSDLs into your integrations. By regenerating the WSDL file, the custom fields of

imported mappings are not deleted, and are available for editing, as needed. This eliminates the need for remapping source and target elements completely from scratch in the mapper.

Restrictions

- Minimize your use of regeneration. Only use this feature when absolutely necessary.
- There cannot be root level differences between the old and new WSDLs. If there are differences, WSDL regeneration fails. Therefore, you can change the connection information, but must ensure that the new WSDL does not have root element differences from the previous WSDL.
- If you create an integration in which the Oracle Service Cloud (RightNow) Adapter WSDL is version 1.2 and try to regenerate it with Oracle RightNow Adapter WSDL version 1.3, a mapper error occurs.
- JCA artifacts are not regenerated.
- A new WSDL can have references to a different schema with different data structures (for example, you change the schema by adding, modifying, or deleting a new complex element to the root element), *only* if the root element within the schema does *not* get modified, added, or removed. In this situation, the child element within the root element can be changed or refer to any new data structure or data type.

  The regeneration logic follows the reference and imports or includes the new data structure to the new WSDL. The root element within the schema is the first level of `<xs:element/>` or type element (`<xs:complexType/>` or `<xs:simpleType/>`). For the first level of `<xs:element/>`, the following applies:

  – Their `QName` cannot be modified.

  – A new root element cannot be added.

  – The old root element cannot be removed.

  For the first level of a type element, the following applies:

  – Their `QName` should not get changed.

  – The old root type should not be removed.

  – A new type element can be introduced by other referencing child elements or types.

  For example, one child element or type defined in a root element can refer to a new type element. The regeneration logic imports or includes them into the existing schema if it has the same target namespace as the child element/type or by creating a new schema if it has a different target namespace from the child element/type. The new root level type element must be referenced by another child element/type. If it is a standalone root type element, it does not get imported or included in the final regeneration WSDL.

> **Note:**
>
> If you receive an error during regeneration, do not save the endpoint/mapping updates. Instead, close and discard the integration without saving, reopen the integration again for editing, and try regeneration again.

To regenerate a WSDL file for integrations:

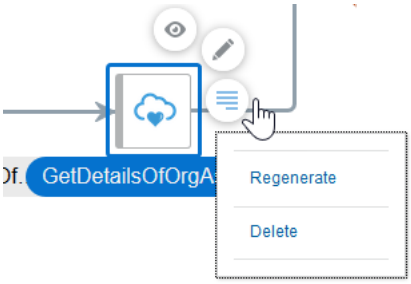
1. In the left navigation pane, click **Home** > **Integrations** > **Integrations**.

2. Click the name of the integration in which to regenerate the WSDL. Customized integrations are designated with the words **BUILT BY ORACLE** and **Customized** to the right of the integration name.

   You can regenerate the WSDL for an individual endpoint or the WSDLs for all endpoints in an integration.

3. To regenerate the WSDL for a single endpoint in the integration, click the appropriate source, target, request enrichment, or response enrichment icon.

   a. Select **Regenerate**.



   You are prompted with a message indicating that WSDL regeneration impacts the mappings in the integration.

   b. Click **Regenerate**.

   This regenerates the WSDL and any dependent artifacts so that any custom elements appear during mapping. The imported mappings from any prebuilt integration are not deleted. The maps are validated and any warnings (identified by yellow icons) or errors (identified by red icons) for the impacted maps are displayed. If warnings and errors both exist for a single mapper, only a single error icon is displayed. Icons indicating that this mapper is customized (identified by the blue icons) are displayed at the bottom of the mapper.

4. To regenerate the WSDLs for all endpoints in the integration, select **Regenerate Endpoints** from the menu.

5. Click **Regenerate** when prompted with a message indicating that WSDL regeneration impacts the mappings in the integration.

   This regenerates the WSDLs and any dependent artifacts with the same behavior.

See Mapping Data in *Using the Oracle Mapper with Oracle Integration Generation 2*.

Modify an Integration

You can modify an existing integration, including changing a source or target connection, reconfiguring the connection, and updating the data mapping. Changes to the source or target can cause changes to the existing mappings.

If the integration you want to modify is active, deactivate it first. See Deactivating an Integration for instructions.

To modify an integration:

1. In the left navigation pane, click **Home** > **Integrations** > **Integrations**.
2. On the Integrations page, find the integration you want to modify.

You can search by entering a partial or complete integration name in the field or filter integrations by selecting an option from the link.

3. Go to the row of the integration you want to change.
4. Click
. You cannot edit an active integration.
5. To modify the trigger or invoke adapter, select the connection in the canvas and click .
6. Modify any necessary fields in the pages of the Adapter Endpoint Configuration Wizard.
7. To assign a new connection as the source or target, click the connection to delete, then click on the menu that is displayed. You cannot delete a trigger connection in an orchestrated integration.
8. Add the new adapter to the canvas.
9. Configure the new connection.
10. To modify a data mapping, click the appropriate map icon and update the mappings. See Mapping Data of *Using the Oracle Mapper with Oracle Integration Generation 2*.
11. When you are done making changes, click **Save** and then click **Close** to return to the Integrations page.

> **Note:**
>
> If you have an integration that is locked and stuck in Draft mode, see Recover Unsaved Integration Changes.

Impact on the Mapper of Editing the Endpoint Information in an Integration

You can edit the endpoint information in an integration that is not active. The changes that you make can impact your mappings. For example, minor edits such as changing the endpoint description do not delete the existing mappings. Major edits such as changing the selected business objects or operations delete the mappings. In these cases, you must recreate your mappings. Before you save your updates, you are prompted to confirm your changes.

The impact of major and minor endpoint changes on an integration are as follows:

- If a minor change is detected, for example:
  - If a map is using either the request or response of the application as a primary input, output, or secondary input, the map is validated.
  - If a map is using a fault of the application as a primary input or output, the map is deleted.
  - If a map is only using a fault of the application as a secondary input, the secondary input is removed.
- If a major change is detected, for example:
  - If a map is using a request, response, or fault of the application as a primary input or output, the map is deleted.
  - If a map is only using a request, response, or fault of the application as a secondary input, the secondary input is removed.

The following are examples of major endpoint changes:

- If the application message exchange pattern changes (for example, from synchronous to asynchronous).
- If a root element name or root element namespace of the input request changes.
- If a root element name or root element namespace of the output response changes.

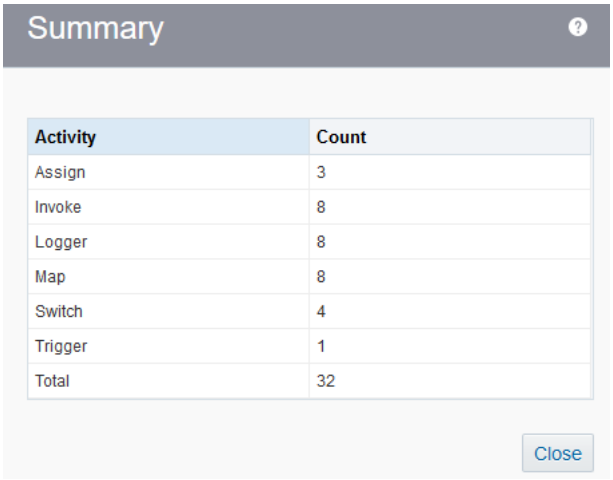
If none of the above changes occur, then the change is considered minor.

1. In the left navigation pane, click **Home** > **Integrations** > **Integrations**.
2. Select the integration to edit.
3. In the integration canvas, select the trigger or invoke endpoint to edit, then click 
.
4. Make appropriate changes in the Adapter Endpoint Configuration Wizard, then click **Done**.
5. Select to confirm your changes when prompted. Minor edits do not delete your mappings. Major edits delete your mappings.

View the Actions and Connections in an Integration

You can view the total number of activities (actions, triggers, and invokes) in an integration.

1. Open an integration.
2. Select **Summary** from the  menu.
A summary or each action, trigger, and invoke in the integration is displayed.



View the Trigger, Invoke, and Enrichment Details of an Integration

You can view the details associated with trigger, invoke, and (if configured for a basic routing integration) enrichment endpoints of an integration.

To view the trigger, invoke, and enrichment details of an integration:

1. In the left navigation pane, click **Home** > **Integrations** > **Integrations**.
2. On the Integrations page, find the integration you want to view. You can filter the display of integrations by their current status with the
   
   link.
3. Open the integration.
4. Select the trigger, invoke, or (if configured for a basic routing integration) enrichment endpoint of the integration.
5. Click the  icon.
6. View the connection name, endpoint name, input payload, and output payload of the integration.

Clone an Integration

Cloning an integration creates a new copy with identical connections and data mappings. You give the clone a new name, identifier, version number, and package name, but the remaining configuration is the same. You can reconfigure the clone after you create it. You can also clone a locked integration.

> **Note:**
>
> Integration versions follow a formatting convention of `xx.yy.zzzz`, where `xx` is the major version and `yy.zzzz` is the minor version. If you clone an integration (for example, version `1.00.0000`) and change the minor version of the cloned integration to `1.10.0000`, version `1.00.0000` is deactivated when you activate version to `1.10.0000`. To keep both integrations active, change the major version of the cloned integration to `2.00.0000`. This enables integration versions `1.00.0000` and `2.00.0000` to be active at the same time. See Create Integrations.

To clone an integration:

1. In the left navigation pane, click **Home** > **Integrations** > **Integrations**.
2. Go to the row of the integration to clone. You can filter the display of integrations by their current status on the left side of the page.
3. Select **Clone** from the
   
   menu.
4. In the dialog that appears, enter a name, unique identifier, version number, package name, and an optional description.

   You can include English alphabetic characters, numbers, underscores, and dashes in the identifier. Enter the version using numbers only in this format: `xx.xx.xxxx`.
5. Click **Clone**.
6. You can modify the clone in any of the ways described in Modifying an Integration.

Create a Draft of an Integration

Creating a draft of an integration creates a new copy with the same integration information. This action represents an easier way to create an integration. During draft creation, you must update the version of the integration and optionally update the package and description. However, unlike a cloned integration, you *cannot* update the integration name or identifier. You can also create a draft of a locked integration.

To create a draft of an integration:

1. In the left navigation pane, click **Home** > **Integrations** > **Integrations**.
2. Go to the row of the integration for which to create a draft. You can filter the display of integrations by their current status with the
   
   link.
3. Select **Create New Version** from the  menu.
4. In the dialog that appears, modify the version number. Not modifying this field results in the following error:

   ```
   The version of Integration "integration_name" you are trying to create
   already exists in the system.
   ```

5. Optionally modify the package name and description. You cannot modify the integration name or identifier.
6. Click **Create Version**.

Delete an Integration

You can delete an integration that is no longer needed.

Make sure the integration you want to delete is not active. To deactivate the integration, see Deactivating an Integration.

To delete an integration:

1. In the navigation pane, click **Home** > **Integrations** > **Integrations**.
2. Go to the row of the integration you want to delete. You can filter the display of integrations by their current status in the
   
   menu.
3. Select **Delete** from the
   
   menu.
4. Click **Delete** when prompted to confirm.

Recover Unsaved Integration Changes

You can recover unsaved changes in an integration that crashes during design time because of a browser crash, loss of network connectivity, a server going down, and so on.

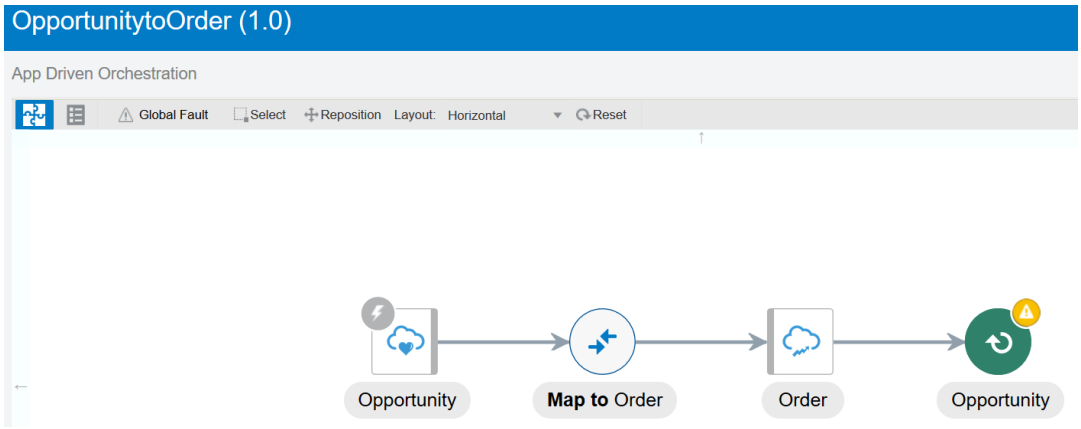
As you design an integration, a backup copy is created whenever you complete a task such as configuring an action, deleting an invoke connection, configuring business identifier tracking, or defining global faults. There is no time limit in which you need to log back in and recover your changes. The unsaved changes remain as long as the integration is not unlocked by an administrator.

The following data in an integration cannot be recovered after a crash:

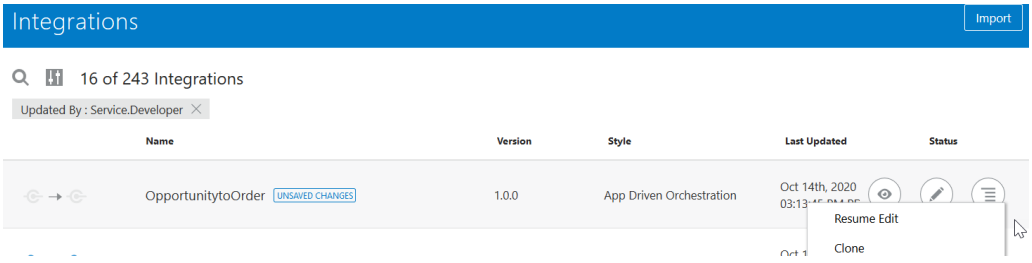
- Changes made while editing an action (for example, you are designing a stage file action when your integration crashes)
- Changes made while working inside the mapper
- Layout changes such as repositioning an action
- Connections page edits
- Any changes in a basic routing integration

Recovery behavior is different based on the roles assigned to the user. For example, assume you are designing the following integration when your browser crashes.

1. Design an integration as a user with the *Service*Developer role. Assume you have not yet saved your changes for the following integration when your browser crashes.

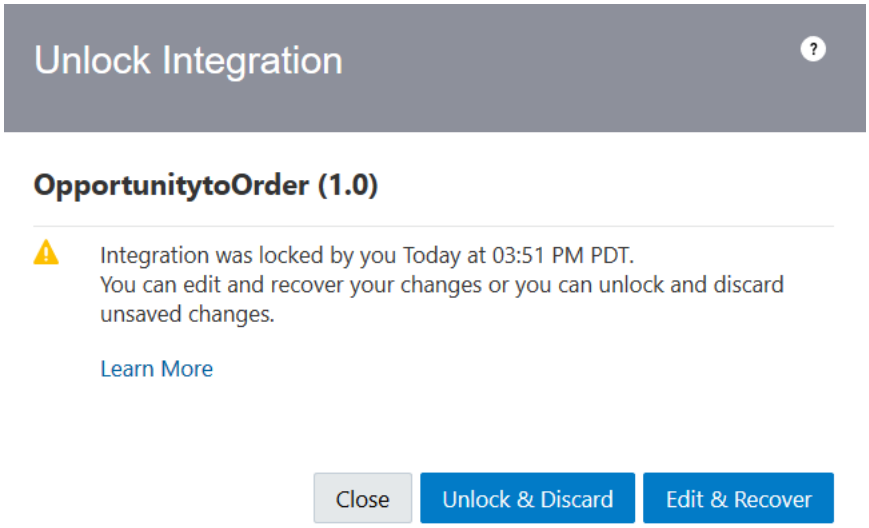


2. If you log back in as that same user with the *Service*Developer role, note the following behavior for recovering your unsaved changes.
   a. Go to the Integrations page and note that the words **Unsaved Changes** appear next to the impacted integration.



b. Select to resume editing and recover your unsaved integration changes in any of the following ways:

- Click the integration name in the **Name** field.
- Click
  
  .
- Select **Resume Edit** from the
  
  menu.
- Select **Unlock** from the
  
  menu, then select **Edit & Recover**.



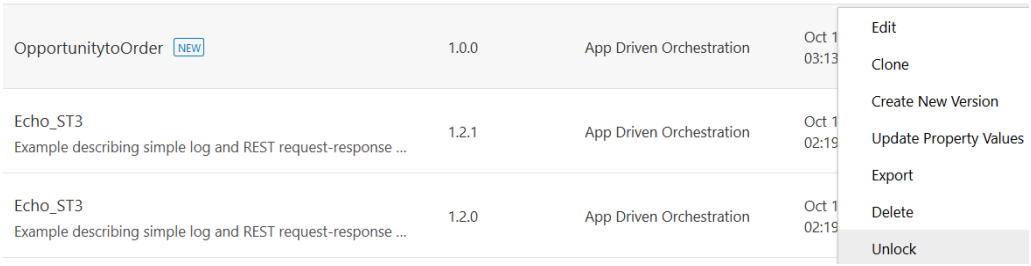
Your integration design is displayed. The following message appears in the banner.

```
Unsaved changes for integration OpportunitytoOrder (1.0) have been
recovered and are displayed in the canvas, but not saved yet.
```

c. Click **Save** to save your changes and continue with integration design.

3. If you instead log in as a user with the *Service*Administrator role, note the following behavior.

a. Go to the Integrations page and note that the word **New** instead of **Unsaved Changes** appears next to the impacted integration.



b. Select from the following options. A user with the *Service*Administrator role cannot recover the unsaved changes. Only the actual designer of the impacted integration can perform that task.

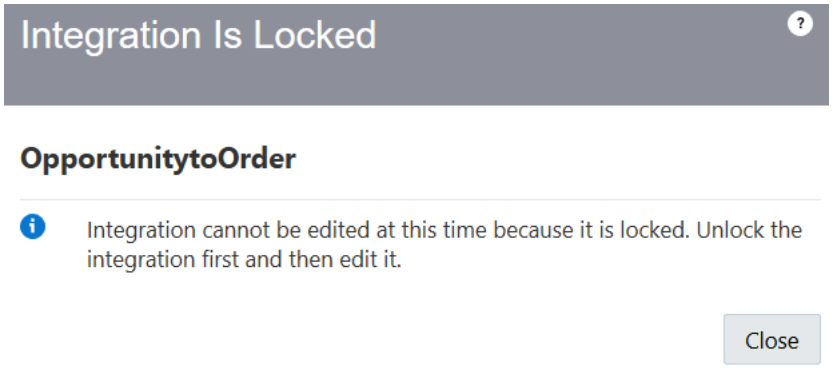
- If you click the integration name in the **Name** field, you see none of the updates you made prior to the crash. The following message is displayed in the banner.

  ```
  Edit is not possible for OpportunitytoOrder (1.0) because this
  integration is locked. Integration is displayed in view mode.
  ```

- If you select **Edit** from the

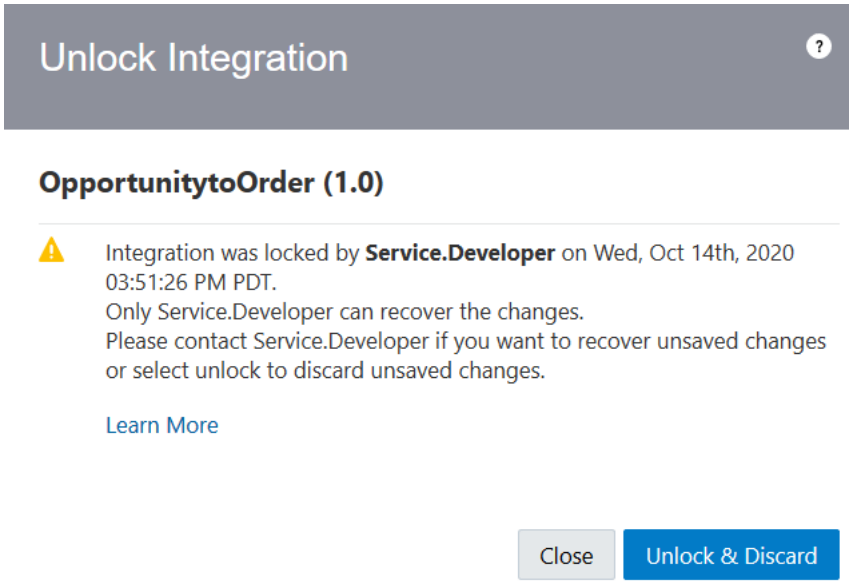
  menu, the following message is displayed.

- If you then select **Unlock** from the

  menu, you are prompted with the following message. If you select **Unlock & Discard**, all changes are lost. Contact the integration designer to determine how to proceed.



Override Design-Time Properties in an Integration

You can configure integration properties outside of the regular integration design. At runtime, these integration properties take effect and override the values you configured at design time. For example, you can specify an email address to use in the **To** field of a notification action during runtime that overrides the email address specified at design-time. This feature enables you to configure the value without having to edit the integration design itself or having to pass the value at runtime.

You can configure all types of properties, including file path, database properties, server properties, credentials, and any constant property that changes at regular intervals.

When you can edit the integration properties is based on the type of integration style you are using.

Integration Style	Editing Behavior
Scheduled	Scheduled integrations support runtime configuration. Therefore, you can update the integration property values when the integration is active. When the scheduled integration is invoked: • If the integration property value is not updated, the default value is used. • If the integration property value is updated with a new value, the new value becomes the current value and is used.

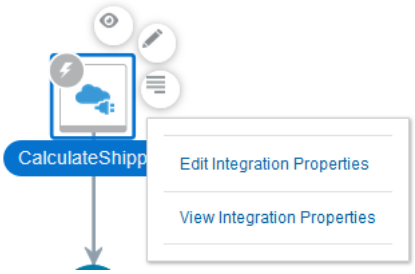
Integration Style	Editing Behavior
App driven orchestration	App driven orchestration integrations do not support runtime configuration. Therefore, you can update the integration property values when the integration is *not* active. The configured property values takes effect after integration activation. When the app driven orchestration integration is invoked: • If the integration property value is not updated, the default value is used. • If the integration property value is updated with a new value, the new value becomes the current value and is used.

Note the following behavior regarding integration properties and major versions (for example, 01.00.0000) and minor versions (for example, 01.10.0000) of an integration.

- The current values of the integration properties are shared by all minor version integrations having the same major version and identifier.
- Updating the current value for an integration property is reflected in all other minor version integrations if they have an integration property with the same name.
- Deleting an integration property from an integration does not remove its current value if there are other minor version integrations with the same major version and identifier.

See Create Integrations for information about versions and identifiers.

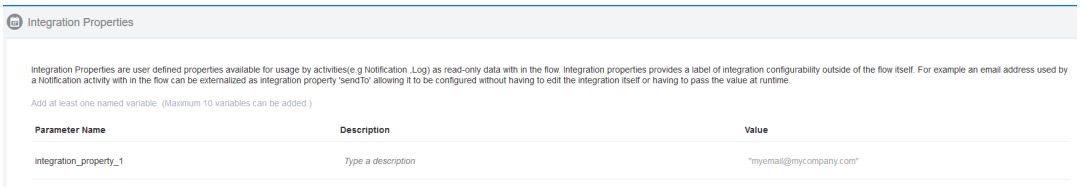
1. Right-click the trigger connection in an integration, and select the **menu** icon.



2. Select **Edit Integration Properties** to add a property.

   The Integration Properties page is displayed.

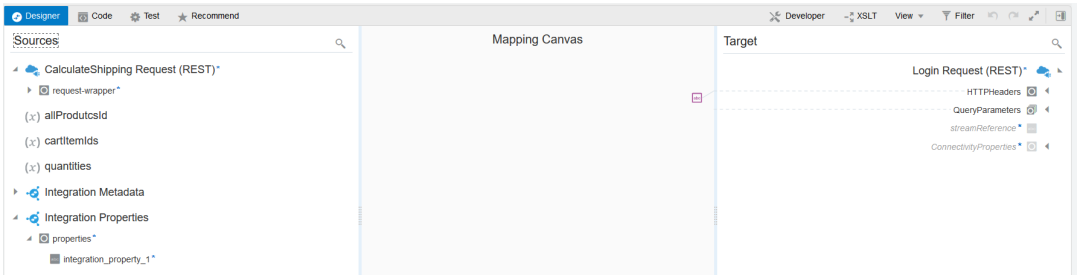
3. Click **+** to add a name, description, and value for the property. For example, enter an email address as the value. A maximum of 10 integration properties can be added.



4. When complete, click **Close** to save your changes and return to the integration canvas.

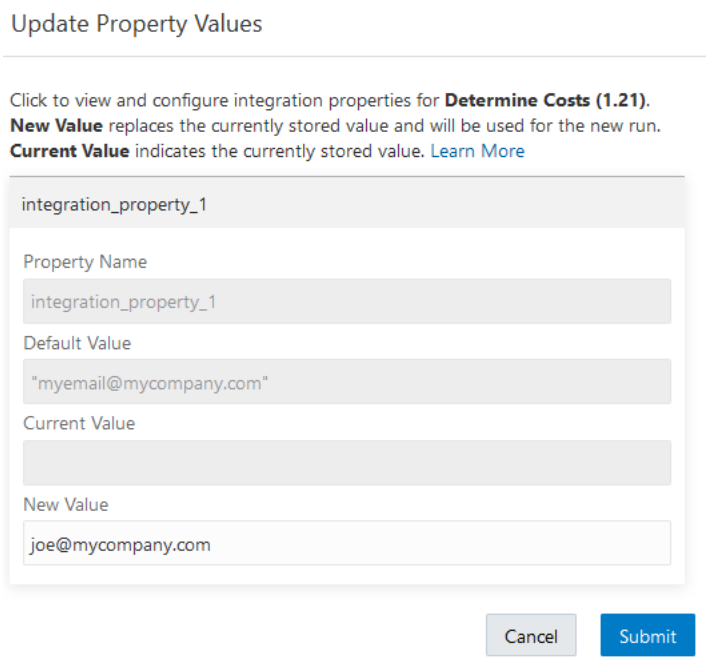
> **Note:**
>
> The integration property value can't be updated within the integration. For example, you cannot update the integration property in an assign action.

5. In the mapper, map the integration property (named **integration_property_1** for this example) visible under **Integration Properties** in the **Sources** element tree to an appropriate element in the **Target** tree.



6. Go to the Integrations page.
7. Find the integration whose properties you want to update, and select .
8. Select **Update Property Values**.

   The Update Property Values dialog slides open on the right side of the page.
9. Click the integration property to update, then enter a value in the **New Value** field and click **Submit**.



A confirmation message indicates that the property was updated.

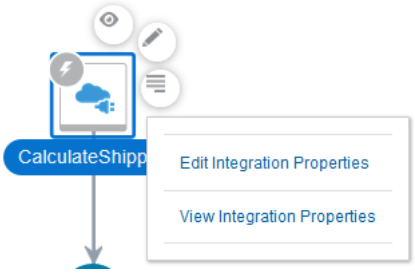
```
Properties for Integration integration_name (integration_version) updated
successfully.
```

The new value is used for a new run of the integration. For this example, the email address you specified in the **New Value** field is used as the **To** field email address of the notification action in the integration.

If you select **Update Property Values** again to return to the Update Property Values dialog, note that the value in the **New Value** field now appears in the **Current Value** field.

You can also view previously configured integration properties.

10. Return to the trigger connections and select **View Integration Properties** to view existing integration properties. This selection provides a read-only view of the properties. The options for adding and deleting existing properties are disabled.



Use Metadata in Integrations

You can access the integration name, identifier, and version instead of hardcoding those values in integrations. You can also access runtime data and environment data inside the integration.

The following metadata is exposed in integrations. This metadata can be used in the mapper and actions that include the Expression Builder such as assign, log, notification, and others.

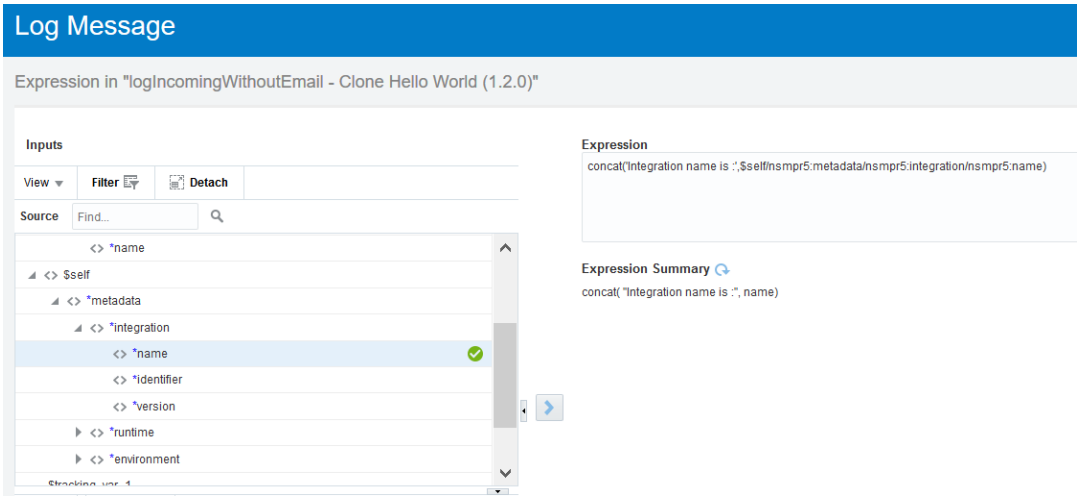
- Integration
  - Name
  - Identifier
  - Version
- Runtime data
  - Instance ID
  - Invoked by name
- Environment data
  - Service instance name
  - Base URL

The following use case provides a high-level overview of using metadata in an integration.

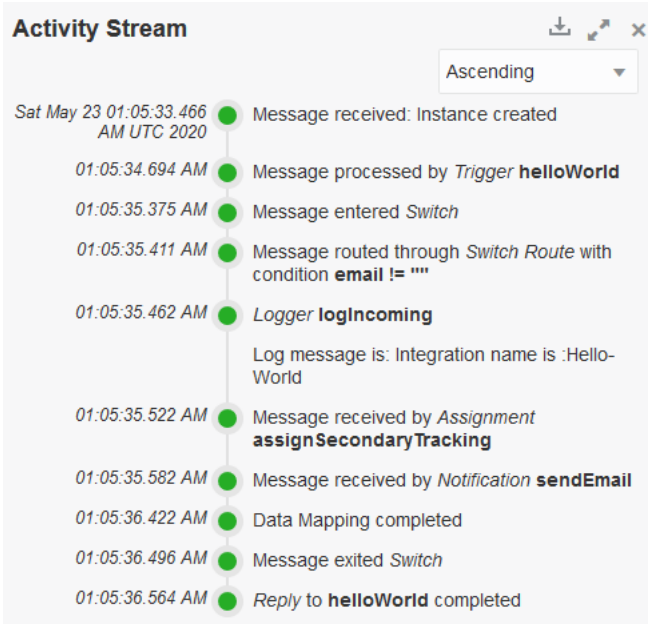
1. Create and design an orchestrated integration.
2. Add an action that includes an Expression Builder.

For this example, a log action is added.

3. In the source tree, find the integration metadata.
4. Add the required metadata to the **Expression** field. For this example, a `concat` function is used to include the integration **name** metadata.



5. Complete design of the integration.
6. Activate and run the integration.
7. Go to the Track Instances page.
8. Click the business identifier of the integration.
9. Select **View Activity Stream** from the menu in the upper right corner.
10. Scroll down and note that the log message in the activity stream shows the integration name (for this example, **Hello-World**).



Cancel a Running Integration Instance

You can cancel an integration instance that is not completing as expected. For example, a looping action such as a while loop does not complete or an agent or cloud adapter connection does not complete.

1. In the left navigation pane, click **Home** > **Monitoring** > **Integrations** > **Tracking**.
2. Go to the row of the integration to cancel.
3. Select
   
   .
   Scheduled instances cannot be canceled from the Track Instances page. Selecting
   
   for a scheduled instance causes a message to be displayed above the banner with a link to the page for canceling the running instance.

View the Contents of an Incomplete Integration

You can open an integration with incomplete connection details on the Integrations page and view its contents. These integrations can only be viewed, and not edited.

To view the contents of an incomplete integration:

1. In the left navigation pane, click **Home** > **Integrations** > **Integrations**.
2. Open an integration with incomplete connection details (for example, an integration imported from another instance or from the Oracle Marketplace).
3. A message is displayed at the top of the page describing why you cannot edit the integration.

   ```
   Edit is not possible for Integration_Name (version_number) because the
   connection settings for this integration are incomplete.
   Update the configuration information for the connections: Connection_Name.
   Integration is displayed in view mode.
   ```

4. Open individual adapters, mappings, actions (for orchestrated integrations), and other components in the integration for viewing.
5. Go to the Connections page for the adapter connection requiring configuration, and update the necessary credential and security details.
6. Return to the integration and note that the message is gone.

Edit an Integration with Incomplete Connections

You can edit integrations that include endpoints based on incomplete connections.

Note the following restrictions and capabilities with editing these types of integrations:

- You can edit completed connections, actions, and maps.
- You cannot edit endpoints based on incomplete connections. Instead, you must first edit the endpoints for those connections on the Connections page.

1. To edit an integration with incomplete connections:

2. In the left navigation pane, click **Home** > **Integrations** > **Integrations**.
3. Open the integration that includes endpoints based on incomplete connections.

   A message is displayed in the banner indicating that you can edit this integration, except for the endpoints based on incomplete connections.
4. Edit any completed connections, actions, and maps, as necessary.
5. To complete the integration, go to the Connections page for the incomplete connection and complete the configuration.
6. Return to the integration and edit the newly completed connection in the Adapter Endpoint Configuration wizard, if necessary.
7. Save and exit the integration, then activate it.

Map Integration Insight Milestones to Integration Actions

Bring real-time visibility and analytics into your integrations using Insight.

As an integral feature of Oracle Integration, Insight simplifies modeling and extracting meaningful business metrics. For background about Insight and its capabilities, see Introduction to Integration Insight in *Using Integration Insight in Oracle Integration Generation 2*.

Mapping Oracle Integration milestones to integration actions requires an understanding of the business process implemented by integration(s). Every time a mapped action occurs in an integration flow, metrics are extracted from the message payload, and the milestone is considered passed. Insight

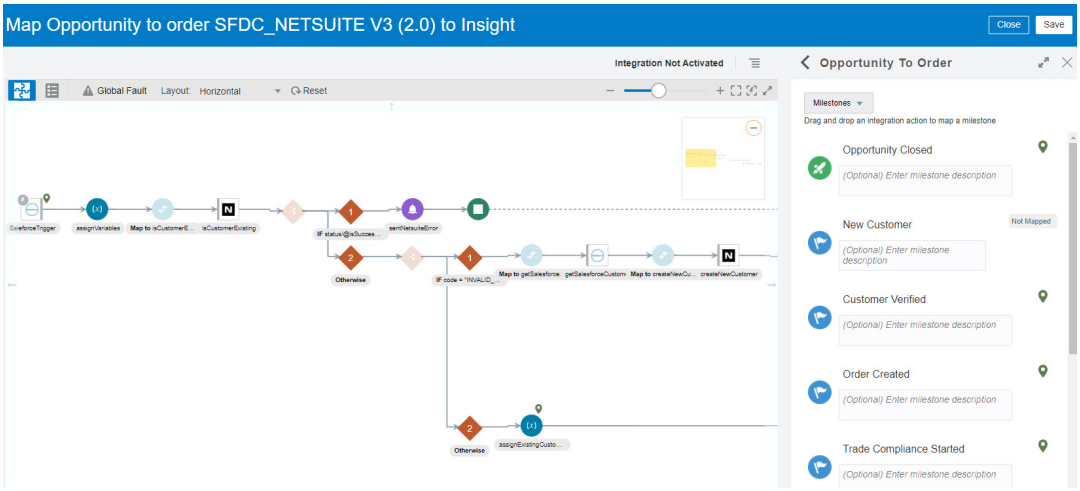
collects this information and displays it on relevant dashboards, allowing business executives to gain insight into their business in real time.
To map Insight milestones to actions in an integration:

1. In the left navigation pane, click **Home** > **Integrations** > **Integrations**.
2. Find the integration to which you want to map milestones. You can filter the display of integrations by their current status on the left side of the page.
3. Hover your cursor over the integration, click

   on the far right and select **Map to Insight**.

   The Insight Designer is displayed.



4. Map the milestones as described in Map Milestones to Integration Actions in *Using Integration Insight in Oracle Integration Generation 2*.

Change the Time Zone

You can change the time zone that is displayed in Oracle Integration.

To change the time zone:

1. In the upper right corner, click the ***username*** icon, then select **Preferences**.
2. From the **Time Zone** list, select the time zone you want to use.
3. Click **Save**.
4. Go to the Integrations page and note that the time zone is changed in the message below the status of the integration and inside the information icon at the far right.

   The time zone change is also shown in other parts of Oracle Integration in which the time is displayed (for example, on the Track Instances page).

Manage an Integration as an API with Oracle API Gateway

After activating a REST Adapter trigger-based integration in Oracle Integration, you can publish the open API specification and deploy the endpoint to Oracle API Gateway.

Note:

- This feature is only available in Oracle Integration Generation 2.
- You cannot configure both Oracle API Gateway and API Platform Cloud Service in Oracle Integration at the same time.
- You can switch between API Platform Cloud Service and Oracle API Gateway, but lose your previous configurations. For example, if API Platform Cloud Service is configured, and you then attempt to configure Oracle API Gateway, you are prompted with a message indicating that your API Platform Cloud Service configurations will be removed if you save your Oracle API Gateway configurations.

Topics:

- Connect to Oracle API Gateway
- Activate and Deploy an Integration to Oracle API Gateway

See Overview of API Gateway and API Gateway Concepts.

Connect to Oracle API Gateway

You must create a connection to Oracle API Gateway before you can activate and deploy an integration from Oracle Integration.

> **Note:**
>
> - The Oracle API Gateway must be in the same region in which your Oracle Integration instance is deployed.
> - You can only connect to Oracle API Gateway in Oracle Integration Generation 2.

1. In the left navigation pane, click **Home** > **Settings** > **Integrations** > **API Management**.
2. Click **API Gateway**.
3. Configure a connection to the Oracle API Gateway. To complete configuration, you must generate an API signing key. See How to Generate an API Signing Key.

   The Oracle Cloud Infrastructure connection must have permissions to deploy to Oracle API Gateway in the necessary compartments as described in the Oracle API Gateway documentation.

   - Course-grained access. See Creating an API Resource with an API Description. For Oracle Integration, the following is required:
     - Access to the compartment with Oracle API Gateway
     - All operations on deployments and descriptions
   - Fine-grained details of permissions. See Details for API Gateway.

Element	Description
Tenancy OCID	Specify the value you copied from the Oracle Cloud Infrastructure Console.
User OCID	Specify the value you copied from the Oracle Cloud Infrastructure Console.
Finger Print	Enter the finger print that was generated when you created the key in the Oracle Cloud Infrastructure.
Private Key	Click  to upload the Oracle Cloud Infrastructure signature V1 or API key details.
Pass Phrase	Enter the pass phrase you created when creating the key.

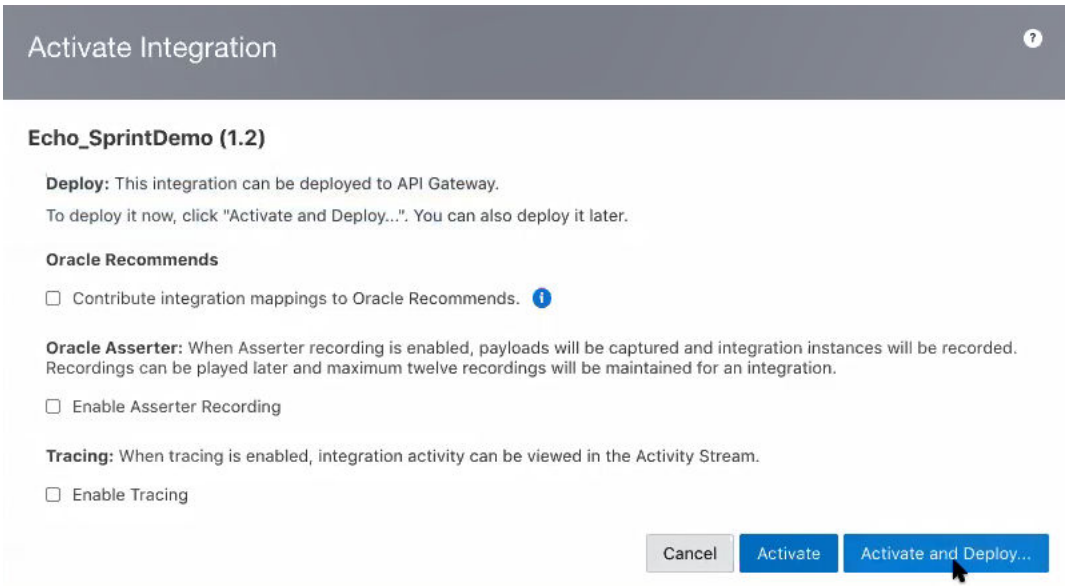
4. Click **Save**.

Activate and Deploy an Integration to Oracle API Gateway

You must activate and deploy your integration to Oracle API Gateway. After completing that action, you can invoke the integration.

1. In the left navigation pane, click **Home** > **Integrations** > **Integrations**.
2. Go to the row of the integration to activate and deploy to Oracle API Gateway. Only REST Adapter triggered-based integrations can be deployed.
3. Click the
   
   icon to activate the integration.

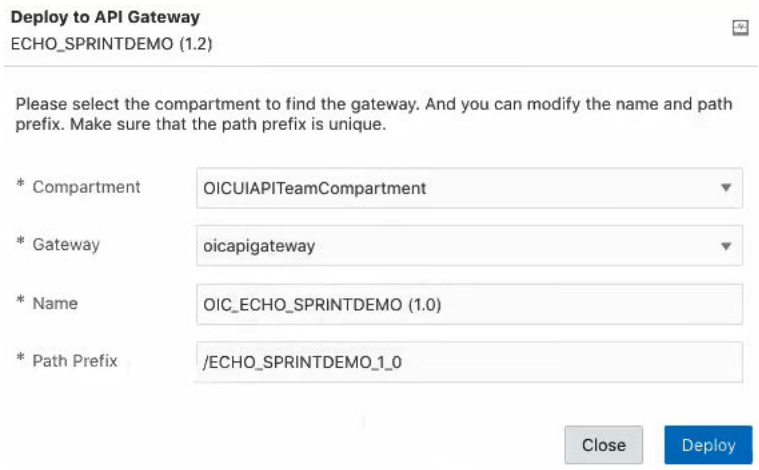
   The Activate Integration page is displayed on the right side. If Oracle API Gateway is configured, the **Activate and Deploy** button is shown.



> **Note:**
>
> - If you deactivate an integration you previously activated and deployed to Oracle API Gateway, the **Activate and Deploy** button is not visible. You can only activate the integration.
> - If you delete an integration deployment to Oracle API Gateway in Oracle Cloud Infrastructure Console, you can create another deployment in Oracle Integration.

4. Click **Activate and Deploy**.

   After activation completes, the Deploy to API Gateway page opens on the right side.



5. Enter the following values:

Element	Description
Compartment	Select the compartment created in the Oracle Cloud Infrastructure Console that includes Oracle API Gateway. **Note**: This list shows a restricted number of compartments in alphabetical order (approximately 100). If your compartment is not listed, then rename it to begin with a letter higher in the alphabet. This action moves the compartment higher in the list.
Gateway	Select the Oracle API Gateway created in that compartment.
Name	Shows the automatically populated display name of the deployment.
Path Prefix	Shows the automatically populated path prefix of the deployment.

6. Click **Deploy**.

   Once complete, the new deployment is created and visible under the specified gateway in the Oracle Cloud Infrastructure Console.

7. Return to the Integrations page in Oracle Integration.

8. From the menu for the integration, select **API Management** to see details about the integration deployment.

9. Go to the gateway to which you deployed your integration in the **API Gateway** section in the Oracle Cloud Infrastructure Console.

10. Copy the URL from the **Endpoint** section under **Deployment Details**.

11. Invoke the deployed integration with the URL (for example, from a browser).

12. Enter your Oracle Integration username and password if prompted.

    The integration is invoked and an instance is created.

13. Go to the Track Instances page in Oracle Integration to view message details.

If you deactivate an integration, any API deployment associated with the integration endpoint is retained in Oracle API Gateway.

Manage an Integration as an API with Oracle API Platform Cloud Service

You can manage integrations as APIs and group several integrations as a single API using the API Platform Cloud Service integration.

Topics:

Create a New API in Oracle API Platform Cloud Service Using an Integration

You can create APIs in API Platform Cloud Service from your integrations with the API Platform Cloud Service integration.

Before you begin:

- Connect to an Oracle API Platform Cloud Service instance. See Connect to Oracle API Platform Cloud Service.
- The integration you want to create an API for must use a REST endpoint as a trigger and must be activated.

To manage an integration as an API:

1. In the navigation pane, click **Home** > **Integrations** > **Integrations**.
2. Go to the row of the integration you want to manage with API Platform Cloud Service.
3. Select **API Platform** from the

menu.
4. From the **Operation** list, select **Create New API**.
5. In the **API Name** field, enter a name for the API.
6. In the **Version** field, enter a version for the API.
7. In the **API Endpoint URL** field, enter the endpoint you want the API to receive requests at when deployed to an API Platform Cloud Service gateway. The API Endpoint URL must be unique for each API deployed to an API Platform Cloud Service gateway.
8. **(Optional)** In the **API Description** field, enter a brief description of your integration. This description is displayed in the API Platform Cloud Service Management and Developer Portals.
9. **(Optional)** To deploy the API to a gateway, select **Deploy API**, and then select the gateways you want to deploy the API to. Only gateways the user (specified in Connect to Oracle API Platform Cloud Service) has deployment rights for are displayed. This option is

grayed out if the user doesn't have the Deploy to Gateway or Request Deployment to Gateway grant for any gateways.

10. **(Optional)** To publish the API to the Developer Portal, select **Publish API**, and then enter a vanity name into the **API Portal URL** field. The vanity name must be unique for each API managed by API Platform Cloud Service. This option is grayed out if the user doesn't have the required grants to publish to the Developer Portal.
11. Click **Create**.

The API is created in the API Platform Cloud Service Management Portal. It is also deployed or published if you chose those options.
After creating the API, you can manage it in the API Platform Cloud Service Management Portal.

Group an Integration in an API in Oracle API Platform Cloud Service

You can group multiple integrations together to manage them as a single API in API Platform Cloud Service.

This makes it easy for you to combine multiple related integrations into a single API in API Platform Cloud Service.

Each integration you group adds a condition to a Resource Based Routing Policy applied to the API in API Platform Cloud Service. When a request is sent to a deployed API requesting the resource you specified for your integration, this policy routes the request to your integration's endpoint (if the request is passed by all policies in the execution flow before the Resource Based Routing policy).

Example: You have two integrations, one for managing orders and another for managing inventory for your pet store. You want to expose these to application developers, but you want to manage who uses them and how many requests are routed to your integrations. To make management and discovery easier, you can group these integrations in a single API in API Platform Cloud Service. When it's deployed, the API will route requests to each integration based on the resource the client requests.

Before you begin:

- Connect to an Oracle API Platform Cloud Service instance. See Connect to Oracle API Platform Cloud Service.
- The integration you want to group must use a REST endpoint as a trigger and must be activated.
- Create an API in API Platform Cloud Service. See Create a New API in Oracle API Platform Cloud Service Using an Integration.

To group an integration in an API:

1. In the navigation pane, click **Home** > **Integrations** > **Integrations**.
2. Go to the row of the integration you want to manage with API Platform Cloud Service.
3. Select **API Platform** from the

menu.
4. From the **Operation** list, select **Add to Existing API**.
5. In the **API Name** field, enter the name of the API to add to the integration.

> **Tip:**
>
> Click the **Search: API Name** icon to view the APIs you can add the integration to.

6. **(Optional)** To deploy the API to a gateway, select **Deploy API**, and then select the gateways you want to deploy the API to. Only gateways for which the user (specified in Connect to Oracle API Platform Cloud Service) has deployment rights are displayed. This option is disabled if the user doesn't have the Deploy to Gateway or Request Deployment to Gateway grant for any gateways.
7. **(Optional)** To publish the API to the Developer Portal, select **Publish API**, then enter a vanity name into the **API Portal URL** field. This option is disabled if the user doesn't have the Manage API grant for the API you selected.
8. Click **Create**.

The integration is grouped with the API.
After grouping an integration with an API, you can manage it in the API Platform Cloud Service Management Portal.

Ungroup an Integration from an API

You can ungroup an integration from an API it was previously grouped in.

When you ungroup an API, the condition for the ungrouped integration is removed from the Resource-Based Routing policy applied to the API. If the API is redeployed if it is deployed to a gateway and republished if it is published to the Developer Portal.

> **Note:**
>
> This task removes an integration that has been grouped with others in an API, described in Group an Integration in an API in Oracle API Platform Cloud Service. Don't use this operation if the integration is the only endpoint added for an API. Delete the API in the API Platform Cloud Service Management Portal instead.

To ungroup an integration from an API:

1. In the navigation pane, click **Home** > **Integrations** > **Integrations**.
2. Go to the row of the integration you want to ungroup from an API.
3. Select **API Platform** from the
   
   menu.
4. In the **Operation** list, ensure that **Remove from Existing API** is selected, then click **Remove**.

The integration is ungrouped from the API. The condition in the Resource Based Routing policy for the integration is removed from the API.

Note:

- If the target API has only one integration grouped to it, ungrouping fails with this error:

  ```
  Remove from existing API failed due to the following reason:
  Operation failed: Unable to remove from a API which contains a
  single endpoint
  ```

- If the API is removed from API Platform Cloud Service, then ungrouping succeeds. You can then create an API from or group this integration to an API as usual.

After ungrouping an integration from an API, you can manage the API in the API Platform Cloud Service Management Portal.

12

Monitor Integrations During Runtime

The Oracle Integration dashboard provides you with the information and tools to monitor and manage your integrations in the runtime environment. Administration tasks can also include working outside the dashboard, such as when you activate or deactivate integrations.

Topics:

Monitor Integrations

Use the Oracle Integration dashboard to see how your integrations are performing. The dashboard provides multiple views for you to check your running services.

Topics:

View the Dashboard

You can view information about how your integrations are performing on the Dashboard page. The Dashboard data represents the entire history of the instance, including the number of messages before purging, if it ever occurred.

You can view information about how your integrations are performing. The main page of the integration shows a snapshot of a state of all running integrations.

To view the dashboard:

1. In the navigation pane, click **Home** > **Monitoring** > **Integrations** > **Dashboard**.

   The Dashboard page provides the following details:

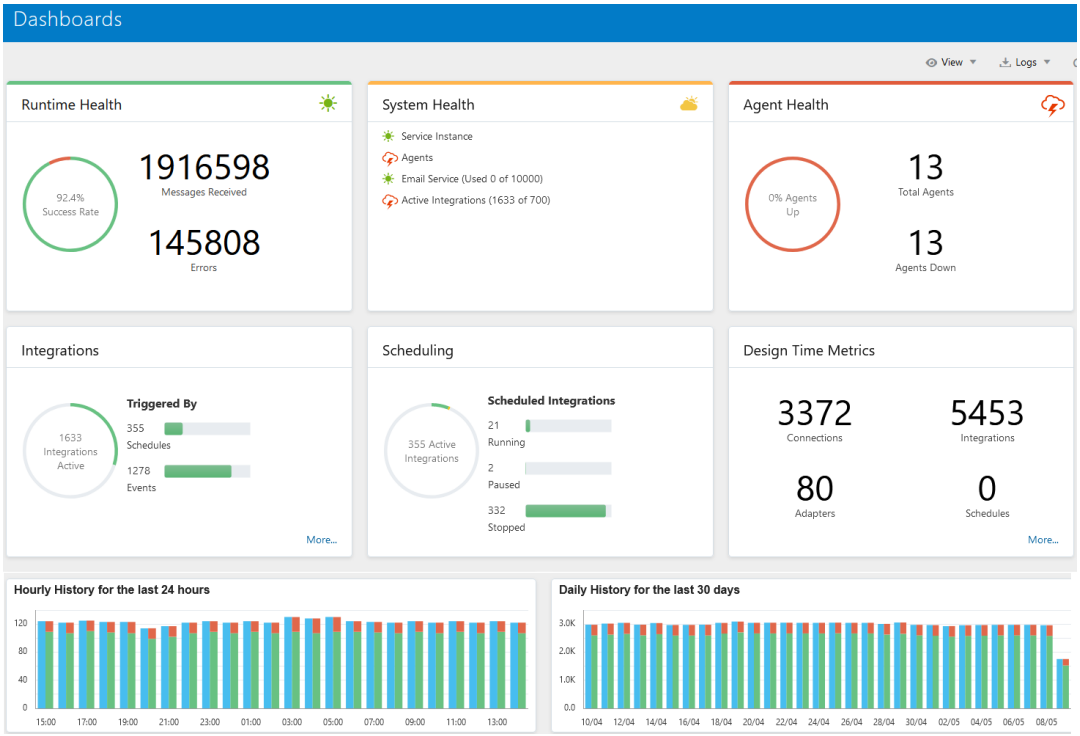
- Success rate for messages (the number of messages received and the number of messages with errors since the Oracle Integration instance was provisioned).
- System health, including the service instance status, agent status, number of emails sent and total number of emails available to send, and number of active integrations. The number of active integrations per Oracle Integration instance cannot exceed 700. Active integrations are defined as currently active integrations and integrations whose activations are in progress. If this number is:
  - above 90% of the limit, a warning message is displayed.
  - equal to or above the limit, a critical message is displayed
- Agent health (the total number of agents versus the number of agents currently not running).
- Number of active integrations and how they were triggered (for example, by schedules or by events).

  Click **More** to access the Monitor Integrations page. See Monitor Integrations.
- Number of scheduled integrations (running and currently paused).
- Design-time metrics showing the number of connections, integrations, adapters, and schedules. Click **More** to access the Design-Time Metrics page. See View the Design-Time Metrics.
- Graphs showing the hourly and daily history of total, successful, and failed instances. You can place your cursor over the bars in each graph to display the total number of successful messages and failed instances.

The Dashboard page reflects historical data. The number of all types of instance states (successful, failed, and so on) are collected hourly and displayed on the Dashboard page. Note the following details about the display of failed and recovered instances:

- Time 1: There are failed instances (shown in the Errors page)
- Time 2: The instances are then recovered (and are now successful)

If instance state information is collected between time 1 and 2, the Dashboard page shows some failed instances. The history graphs on the Dashboard page also show the hour/day during which that was the instance state. If the recovery of failed instances occurs before any new data is collected, the Dashboard page shows those instances as successful.



2. From the **View** menu, select an option:
   - Select **Activity Stream** to view the activity stream and message activity of invoked integrations. See View the Activity Stream from the Dashboard Page.
   - Select **Design Time Audit** to view the design-time audit of all integrations. See View the Design-Time Audit.
3. From the **Logs** menu, select the type of log to download. This menu is *not* available if you are using Oracle Integration Generation 2.
   - **Diagnostic Logs**: If you are having problems with an integration, you can attach the diagnostic logs to a service request for help in debugging the issue.
   - **Incident Logs**: Click to download the created incident report. See Report Incidents.

> **Note:**
>
> The ability to download incident logs has been deprecated.

View the Activity Stream from the Dashboard Page

You can view the activity stream and message activity of invoked integrations. The activity stream provides status about the payload as it moves through the integration during runtime, including any failures.

> **Note:**
>
> - During periods of high loads, expect a delay in the display of instance activity data. This is because the data in the log file may not be available. The logging action in orchestrated integrations may take some time to write the data into the log file. The amount of time it takes to write the data to the log file is based on the size of the load.
> - After activation of an integration, there is a several second delay between when the first instance is created and when the message is processed in the activity stream. This is the expected behavior. See Delay in Message Processing of First Instance after Activation.
> - The supported total size of the activity stream is restricted to less than 100 MB.
>
> The maximum size for payloads inside the activity stream is 512 KB, at which time the payloads are truncated.
>
> See Activity Stream Logs Deleted.

To view the activity stream from the Dashboard page:

1. From the **View** menu, select **Activity Stream**. The Activity Stream page shows details about the activity stream of invoked integrations.
2. Click **Download** to download the activity stream.

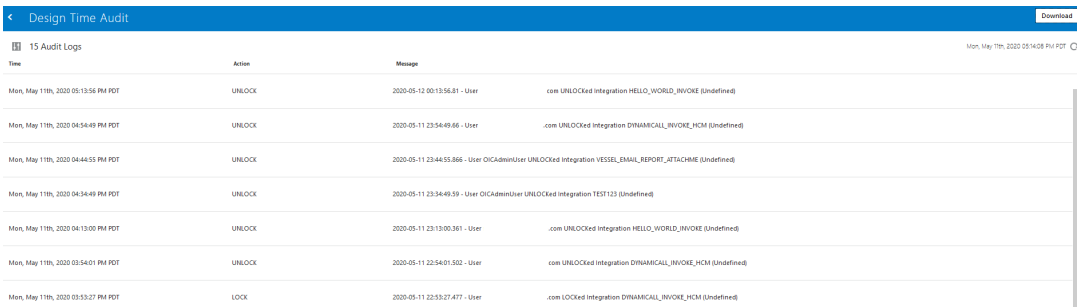
View the Design-Time Audit

You can view the design-time audit of all integrations.

To view the design-time audit:

1. From the **View** menu, select **Design Time Audit**.

   The Design Time Audit page shows details about the audit for all invoked integrations.



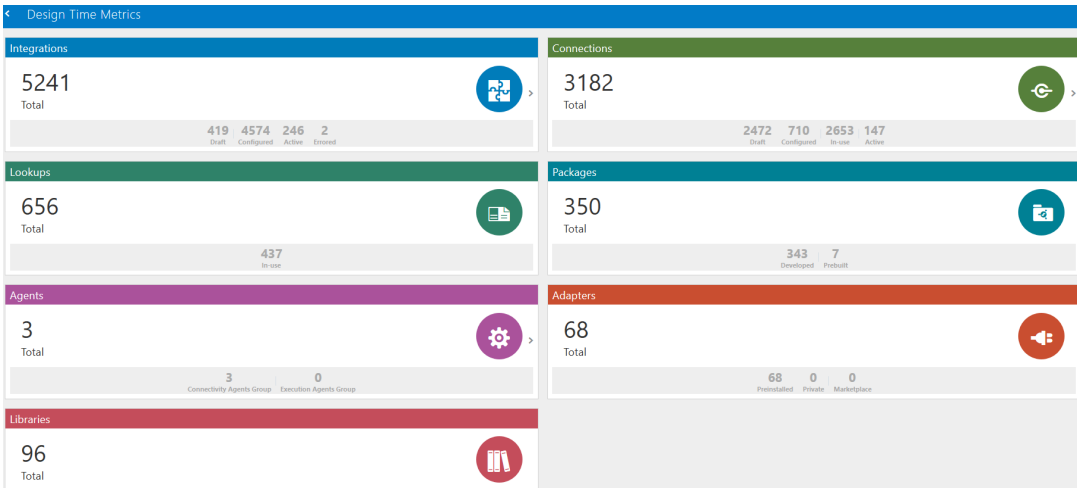
2. Click **Download** to download the design-time audit.

View the Design-Time Metrics

You can view the design-time metrics of Oracle Integration, including details about integrations, connections, lookups, packages, agents, and adapters.

To view the design-time metrics of Oracle Integration:

1. In the navigation pane, click **Home** > **Monitoring** > **Integrations** > **Dashboard**.
2. In the **Design Time Metrics** box, view the number of connections, integrations, adapters, and schedules.
3. Click the **More** link.
4. View details about the following components in Oracle Integration:
   - The state of integrations (draft (under construction), configured, active, and failed).
   - The state of connections (draft, configured, in use, and active).
   - The number of lookups, including in use.
   - The number of packages (developed and prebuilt).
   - The number of connectivity agents.
   - The number of adapters (preinstalled, private, and marketplace).
   - The number of libraries (draft and configured).



5. Click the **>** arrow for **Integrations** to display details about the number of integrations configured for the following:
   - Application-driven integrations
   - Basic routing integrations
   - Publish to OIC
   - Subscribe to OIC
   - Scheduled orchestrations
6. Click the **>** arrow for **Integrations** again to view how many integrations are using each adapter.

7. Click the **>** arrows for **Connections** and **Agents** to view more specific details about the configured adapters and configured agents (execution and connectivity).

Monitor the Number of Emails Sent from Oracle Integration

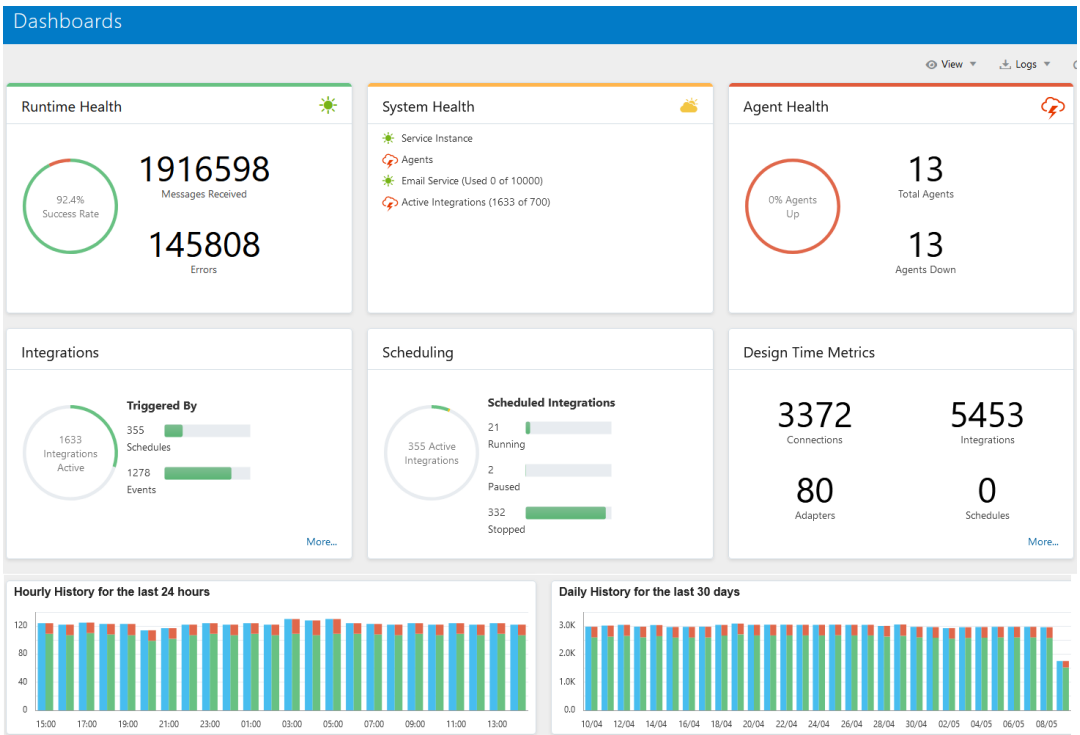
You can send up to 10,000 emails daily from Oracle Integration. When the limit is exceeded, email delivery occurs successfully one last time. For the next (second) delivery, emails stop being sent and a `Daily outbound email limit crossed` error message is written to the activity stream for the impacted integration instance. You can monitor the current email count on the Dashboard.

To handle this type of scenario in your integration, you can also include the notification action in a scope action so that you can emit log messages in a fault handler and then resume instance execution, if required.

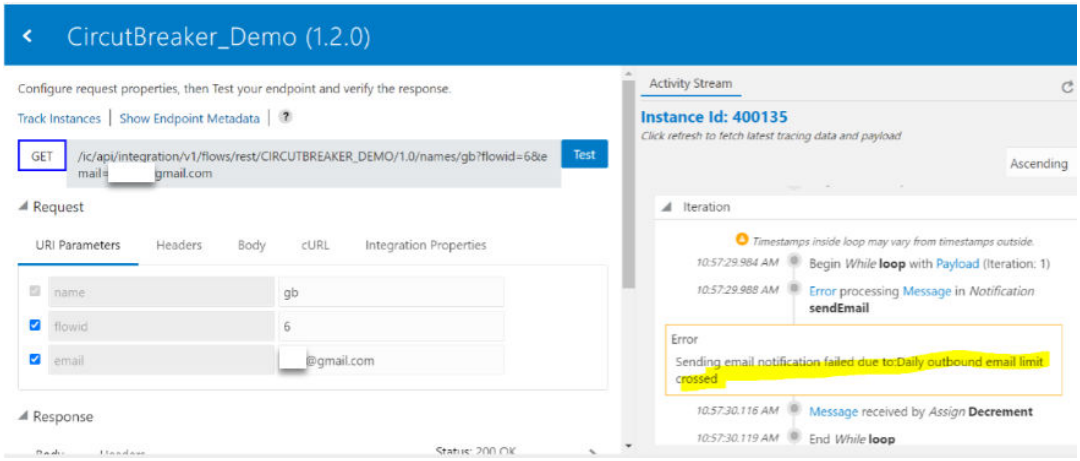
Note the following:

- The email quota is built on a 24 hour rolling window. At any particular point in time, the last 24 hours of outbound emails are counted.
- When you exceed your quota in only one hour (same hour), you are blocked from sending emails for the next 24 hours.

1. Invoke integrations that use the notification action.
2. In the navigation pane, click **Home** > **Monitoring** > **Integrations** > **Dashboard**.
3. Monitor message delivery quota status from the **System Health** section of the Dashboard page. For this example, the daily quota is set to `10000` (the default value) and `0` messages have been sent (quota not yet exceeded).



4. Continue to invoke integrations. For this example, a REST Adapter trigger-based integration is invoked from the Test page. See Test REST Adapter Trigger Connection-Based Integrations.
5. View the activity stream for the integration instances. When the email quota is exceeded for a second time, an error message is displayed in the activity stream. For example:



Monitor Integrations

The Monitor Integrations page enables you to view the message processing status of your running integrations, such as how many messages have been received and processed, how many successful messages and errors have occurred, and how many messages have been aborted. Only activated integrations are listed on this page by default. Using the appropriate filter, you can also view data for inactive integrations.

A **Total Instance Count** summary is displayed at the top of the page. This section is collapsed by default. Data is not immediately fetched. To fetch and view data, expand the **Total Instance Count** section and click the **Refresh** button visible within the section.

This summary shows the count of total available messages in various states present in the tracking store. This summary is different from the metrics shown on the Dashboard page. While the Dashboard page metrics consist of historical data collected since the environment was created, the Monitor Integrations page shows only the current set of messages that have not yet been purged.

To monitor integrations:

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

   A list of active integrations appears by default, along with processing information about the number of messages received, the number of messages processed, the number of successful messages, the number of failed messages, and the number of aborted messages. For scheduled integrations, navigation links are provided that take you to the Schedule and Future Runs page.
2. From the

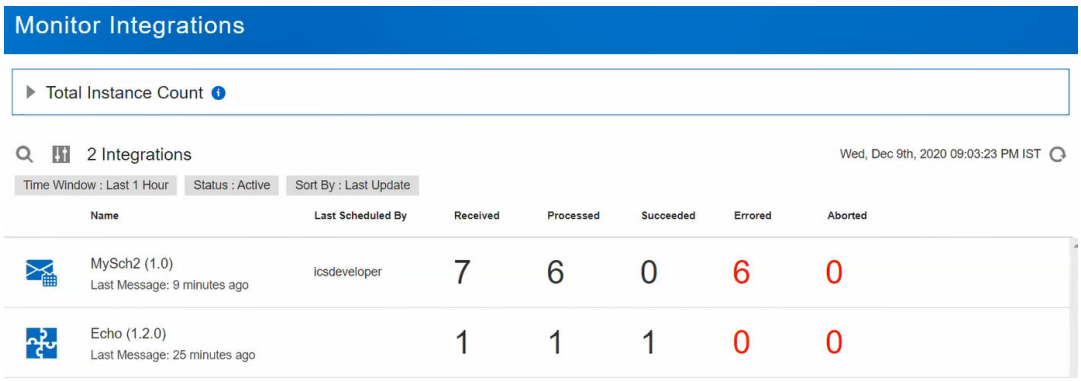
   link, select the time window, integration status, or integration style for which to search for integrations.

   You can also specify a custom time range by clicking

   to filter the search for integrations with the **From** and **To** buttons at the bottom. This

functionality also exists on other monitoring pages such as the Track Instances or Errors page. See Step 3 of Track Business Identifiers in Integrations During Runtime for details.



The user who last submitted the schedule of an integration is displayed in the **Last Scheduled By** column. For non-scheduled integrations such as app driven integrations or basic routing integrations, this column remains empty.

The values shown in the table columns are based on the selected time window (last 1 hour, last 6 hours, and so on):

- **Received**: Shows the number of new incoming messages (or instances created). It is possible that some of these messages will not complete processing within the selected time window.
- **Processed**: Shows the sum of the **Succeeded**, **Errored**. and **Aborted** column values. This is the number of messages that completed processing in the selected time window. Some messages may have been received in a prior time window, but completed processing during the currently-selected window.
- **Succeeded**, **Errored**, and **Aborted**: The explanation is the same as the **Processed** column count. The only difference is the different terminal states.

A difference can exist between the number of messages listed in the **Received** and **Processed** columns. For example, assume there are 76 **Received** column messages and 184 **Processed** column messages. This means that for the selected time window, 76 new messages were received. In that same time window, 184 messages completed processing. Many of the 184 would have been received before the selected time window, and therefore not counted in the **Received** message total. Similarly, many of the 76 **Received** messages may not complete within the selected time window. Therefore, even though they are counted in the **Received** total, they are not counted in the **Processed** total.

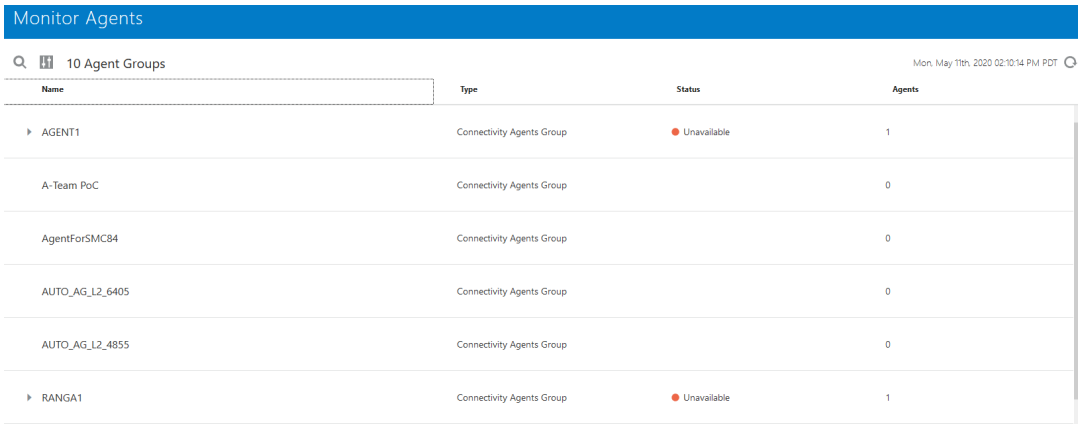
3. Click numbers in the **Received**, **Processed**, **Succeeded**, and **Aborted** columns to access the Track Instances page.
4. Click numbers in the **Errored** column to access the Errors page.
5. For scheduled integrations (identified by ), you can perform the following tasks:
   a. Click
   
   to submit the integration now. If you are a user with the ServiceAdministrator role, you can change the user with which to submit the schedule.
   b. Click
   
   to schedule an integration or pause a schedule.

Monitor Agents

You can monitor the agent groups and their associated connectivity agents in Oracle Integration.

To monitor a connectivity agent:

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



2. The Agent Monitoring page shows details such as the time at which the agents were last updated and the connectivity agents associated with the agent groups.
   - If the agent status is green, this indicates that the agent is running and able to process messages.
   - If the agent status is red, you cannot select it in the Select an Agent Group dialog of the Connections page, even though the agent may be up and running. Review the `agent-diagnostic0.log` file under `agenthome/logs` for more detailed information about errors.
3. Expand the agent group name to see the connectivity agent associated with this group.
4. Click
   
   to also show details about the agent group such as creation date and the last startup date of the connectivity agent.

> **Note:**
>
> When you shut down the agent server, several minutes pass before the agent is displayed as down on the Agent Monitoring page (as indicated by a red thunderbolt).

Manage Errors

You can manage errors from the Errors pages in Oracle Integration. You can resubmit failed messages, discard errors, view message recovery status, view error messages, and view

specific error details, including the activity stream, business identifiers and values, and message payload in failed instances.

Topics:

- View Specific Error Details
- Resubmit Failed Messages
- Abort Errors
- View the Status of Message Recovery
- View Errors By Integration and Connection
- View the Activity Stream of Failed Integration Instances
- View Business Identifiers in Failed Integration Instances

See About Error Management.

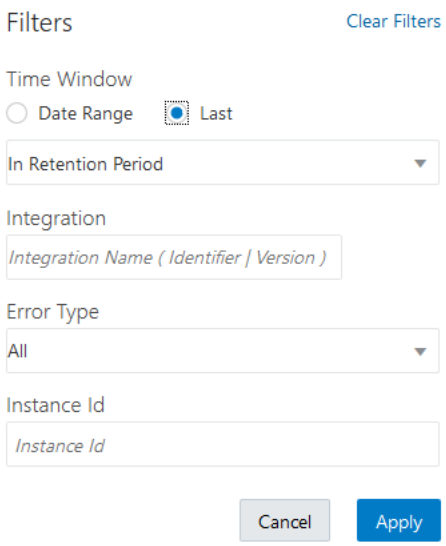
View Specific Error Details

You can view specific error details by integration instance.

Error message details can be displayed in two parts: a summarized error message for easy understanding and resolution and, if necessary, a more specific error message if more detailed troubleshooting is required.

To view specific error details:

1. In the left navigation pane, click **Home** > **Monitoring** > **Integrations** > **Errors**.
2. Click
   
   to filter the display of errors by time window, integration, error type (all, recoverable, nonrecoverable, or recovery job) and instance ID, as needed.



3. Click **Date Range** in the **Time Window** section to specify a custom time range with which to filter the search for errors with the **From** and **To** buttons. This functionality also exists on other monitoring pages such as the Track Instances or Monitor Integrations page. See Step 3 of Track Business Identifiers in Integrations During Runtime for details.

4. View details about a specific error in either of two ways:

   a. Click

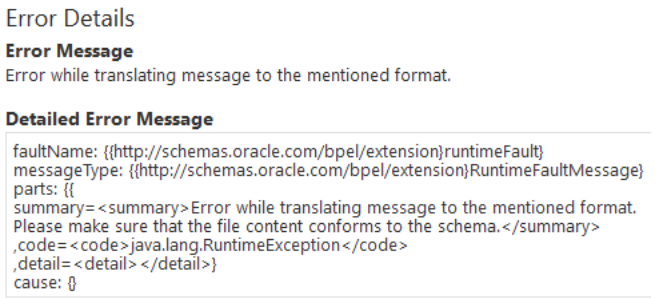
   in the row of the failed instance.

   A summarized version of the error message activity stream for easy understanding and resolution is displayed.

   b. If a more detailed description of the error message is required, click

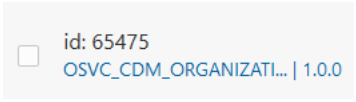
   .



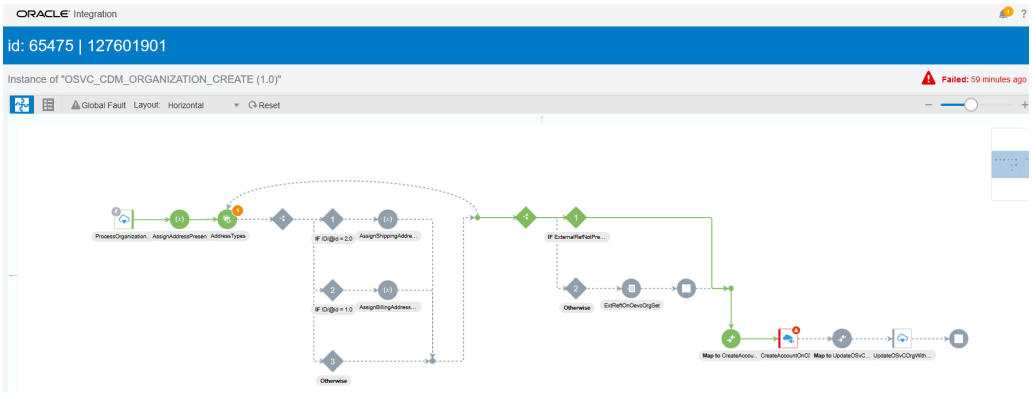
   .

or

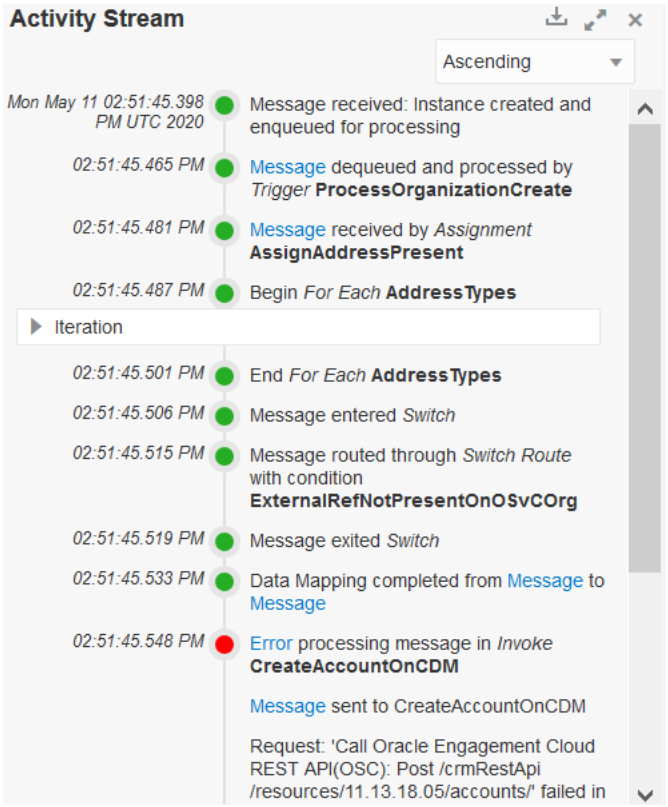
a. Click the business identifier ID included in the failed instance (for this example, **65475**).



The failed integration instance is displayed.



b. In the upper right corner, click  > **View Activity Stream**.



.

Resubmit Failed Messages

You can manually resubmit failed messages. If a local scope has been defined in the integration, resubmission starts at the local scope level. Otherwise, resubmission starts from the beginning of the integration. Oracle Integration does not automatically resubmit failed messages.

All faulted instances in asynchronous flows in Oracle Integration are recoverable and can be resubmitted. Synchronous flows cannot be resubmitted. You can resubmit errors in the following ways:

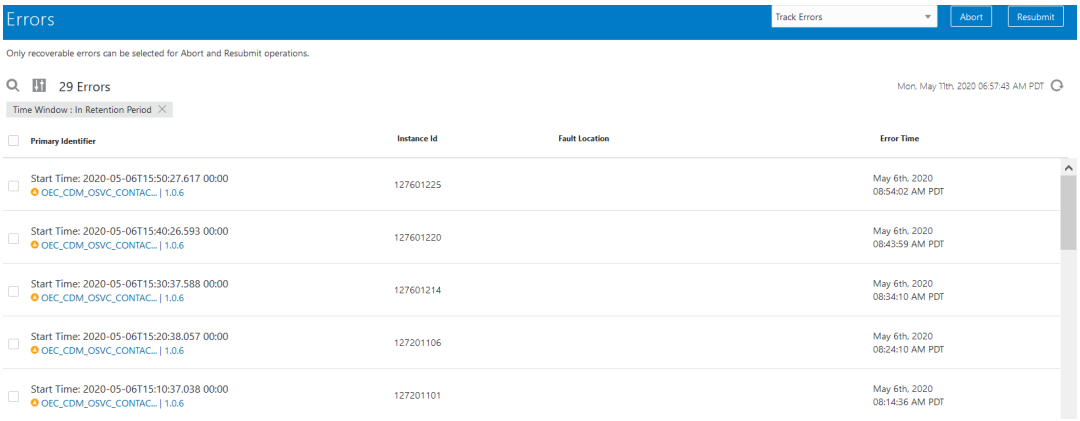
- Single failed message resubmissions
- Bulk failed message resubmissions

Error instances that are resubmitted and successfully resolved are removed from the error list. If an instance is resubmitted and is in progress, a state of `In Progress` is displayed in the list. During this state, additional resubmits of this error instance are not permitted.

> **Note:**
>
> Do *not* discard a message that you want to resubmit. A discarded message cannot be resubmitted.

To resubmit failed messages:

1. In the left navigation pane, click **Home** > **Monitoring** > **Integrations** > **Errors**.
2. From the link, select the time period or retention period during which to search for integration errors.



3. Resubmit errors in either of two ways:
   a. Select the check boxes of errors to resubmit together.
   b. Click **Resubmit** in the upper right corner.

   or

   a. Go to the row of the specific error to resubmit.

b. Click

.

A message is displayed at the top of the page:

```
Instance ID "number" is submitted for processing. Click on the refresh
button to see if it still appears on the errors list before another
resubmission.
```

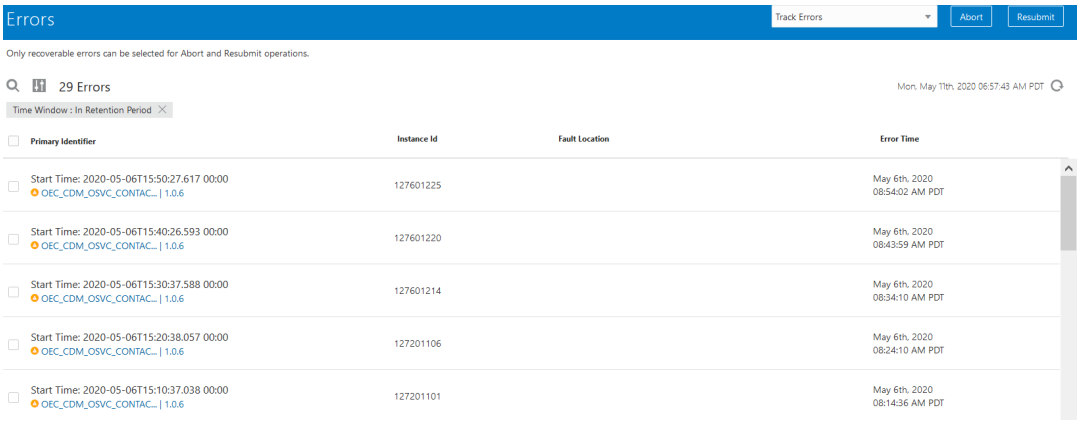
Abort Errors

You can abort errors based on the integration in which they occurred. A aborted error message is removed from the Errors page and can be seen in an aborted state on the Track Instances page. You cannot perform any further operations on a aborted message, including recovery. After a certain time period, the error message is permanently deleted from the server.

To abort errors:

1. In the left navigation pane, click **Home** > **Monitoring** > **Integrations** > **Errors**.
2. From the  link, select the time period or retention period during which to search for integration errors.

   You can also specify a custom time range for which to search for errors.



3. Abort errors in either of two ways:
   a. Select the check boxes of individual errors to abort.
   b. Click **Abort** in the upper right corner.

   or

   a. Go to the row of the specific error to abort.
   b. Click  .

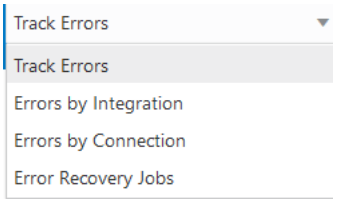
A message is displayed at the top of the page:

```
Instance ID "number" is submitted for abort. Click on the refresh button to
see if this instance still appears on the errors list before trying to abort
again.
```

View the Status of Message Recovery

You can search for and view the status of failed messages that have been submitted for recovery on the Errors page.

1. In the left navigation pane, click **Home** > **Monitoring** > **Integrations** > **Errors**.
2. Submit a failed message for recovery.
3. From the dropdown list at the top right, select **Error Recovery Jobs**.

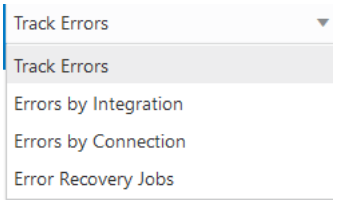


4. The Error Recovery Jobs page is displayed for viewing the status of message recovery.
   - **Received**: Recovery is still in progress.
   - **Succeeded**: Recovery succeeded.
   - **Errored**: Recovery failed.

View Errors By Integration and Connection

You can view errors by integration and connection on the Errors page.

1. In the left navigation pane, click **Home** > **Monitoring** > **Integrations** > **Errors**.
2. To view errors by integration, select **Errors by Integration** from the dropdown list at the top right.



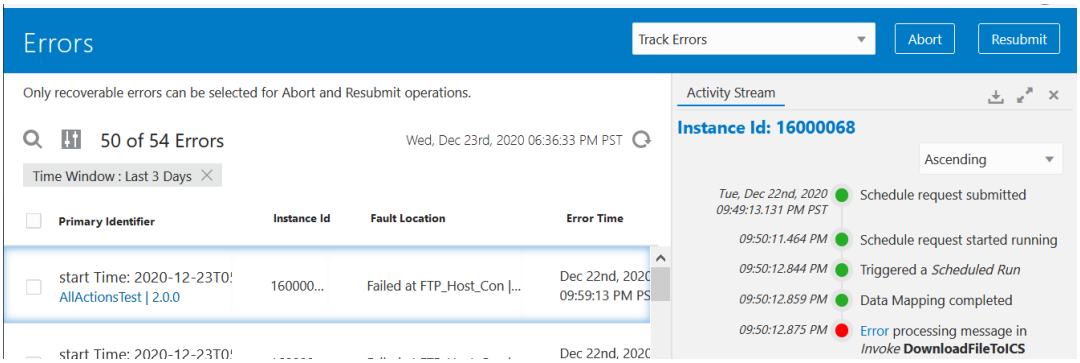
   a. Click
   
   to resubmit errors.
   b. Click  to abort errors.
3. To view errors by connection, select **Errors by Connection** from the dropdown list at the top right.
   a. Click
   
   to resubmit errors.
   b. Click  to abort errors.

View the Activity Stream of Failed Integration Instances

You can view the activity stream of a failed integration instance. This enables you to see where an integration error occurred in the message flow.

To view the activity stream of a failed integration instance:

1. In the left navigation pane, click **Home** > **Monitoring** > **Integrations** > **Errors**.
2. Click
   
   at the end of the row of a specific failed instance to view the message flow. The activity stream shows details about the movement of the message through the integration, including where any failures occurred. For orchestrated integrations, the message flow through each action (for example, any defined switch action branches) is also shown. The date and time according to your user preferences are displayed.



   The functionality available is similar to that on the Track Instances page. See Track Business Identifiers in Integrations During Runtime.
3. Click the business identifier included in the instance you want to view.

   The instance integration is displayed. The direction in which the error occurred is indicated by the color red.
4. In the upper right corner, click  > **View Activity Stream**.

   The activity stream shows details about the movement of the message through the integration, including where the failure occurred.

View Business Identifiers in Failed Integration Instances

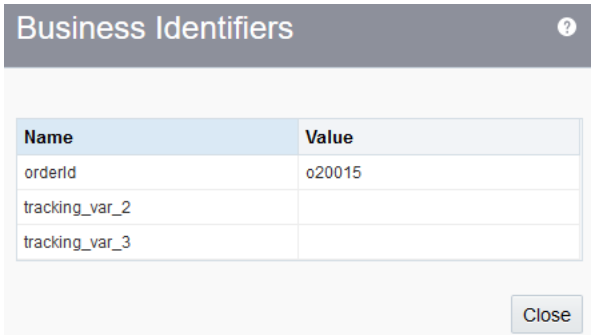
You can view the business identifiers included in failed integration instances.

To view business identifiers in failed integration instances:

1. In the left navigation pane, click **Home** > **Monitoring** > **Integrations** > **Errors**.
2. Click the business identifier of the failed integration instance.

   The integration instance is displayed.

3. Click and select **Business Identifiers** to display all the defined business identifiers and values in the integration.



See Assign Business Identifiers and Manage Business Identifiers for Tracking Fields in Messages.

Manage Business Identifiers for Tracking Fields in Messages

You can view the status of business identifiers included in integrations on the Tracking page.

You can also view the message payload of an instance you are tracking.

Topics:

- Track Business Identifiers in Integrations During Runtime

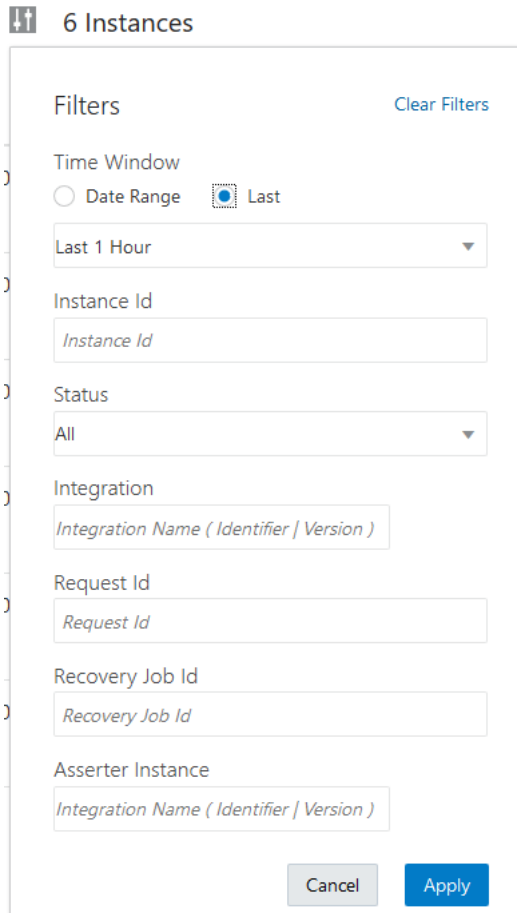
See Assign Business Identifiers for Tracking Fields in Messages.

Track Business Identifiers in Integrations During Runtime

You can track fields in messages on which you have defined business identifiers on the Track Instances page during runtime. These fields are only available for tracking on the Track Instances page if you defined a primary business identifier in the Business Identifiers for Tracking dialog during design time.

To track business identifiers in integrations during runtime:

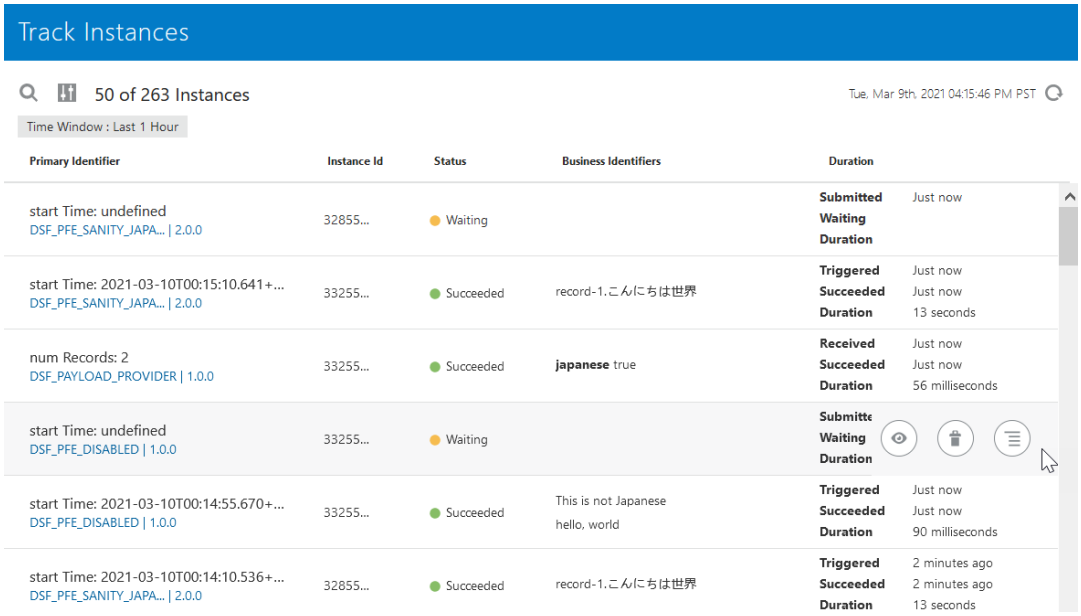
1. In the left navigation pane, click **Home** > **Monitoring** > **Integrations** > **Tracking**.
2. From the
   
   link, select the time window, instance ID, status, integration, request ID, recovery job ID, or asserter instance for which to search for business identifiers in messages.



You can also click **Date Range** to specify a custom time range with which to filter the search for instances with the **From** and **To** buttons at the bottom. This enables you to select a time range as small as a minute and track only those instances processed within that specified time range. The specified time range is preserved by default and applied to the data being displayed when you navigate from one monitoring page to another (for example, when you move from the Track Instances page to the Monitor Integrations or Errors page).

> **Note:**
>
> The time range specified must be within the retention period. Data beyond the retention period is not retained. If the selected time range is beyond the retention period, only available data is displayed.

Results are displayed for any integration on which a primary business identifier is set, including the business identifier and value, the instance identifier of the integration, and the state of the integration (for example, completed, failed, or aborted). You can show the name and value of tracking variables.
For aborted instances, the running time does *not* point to the actual time the aborted (faulted) instance was running. Instead, it consists of the total time span between when the instance began and when the instance was manually aborted.



3. Use the **Search** field to search for values across both primary identifiers and secondary identifiers.

   a. To search for primary and secondary identifier values, enter a value, and click **Search**. For example, assume the primary value is `James`. In the search results, any instances that have a primary value of **James** are returned. Any instances with a secondary value that includes **James** (for example, **Mark.James@asc.com**) are also returned.

   b. To search only on primary business identifier values, enter the value as `primary: value` or `Primary: value` in the **Search** field

> **Note:**
>
> - If you enter a primary business identifier in the **Search** field, but do not click the **Search** button, then select a value from the time period dropdown list, note that the instances are filtered considering the string entered in the **Search** field, even though the **Search** button was not clicked. This is the expected behavior and is true for other landing pages in Oracle Integration.
> - The search facility on the Track Instances page is case sensitive.

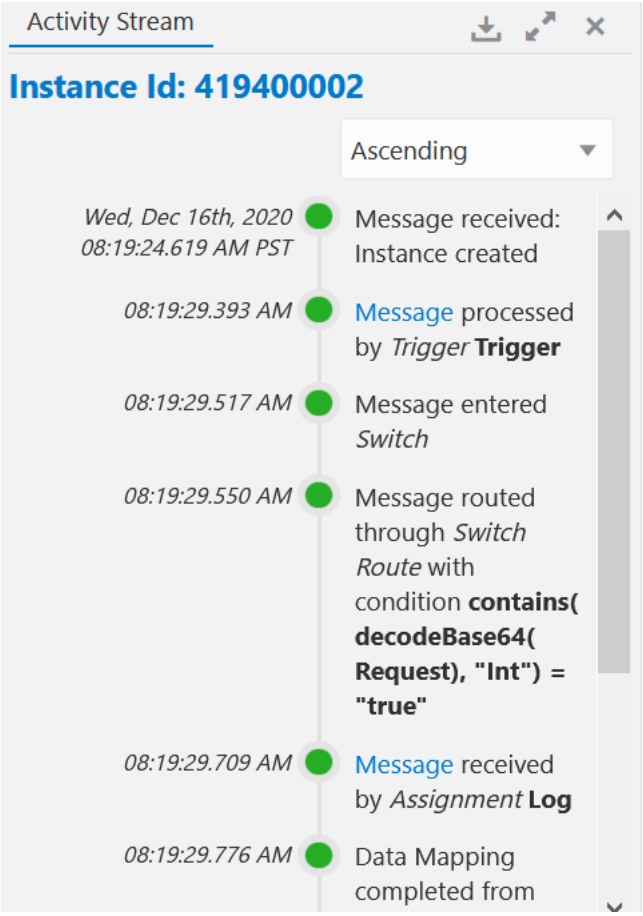
4. Click  at the end of the row of a specific instance to abort it. This option is only available under the following scenarios:

   - In-progress instances of asynchronous integrations (both scheduled and nonscheduled).

- Waiting and paused instances of scheduled integrations. Attempting to abort other types of instances results in an error message being displayed.

5. Click

at the end of the row of a specific instance to view the message flow. The activity stream shows details about the movement of the message through the integration, including where any failures occurred. For orchestrated integrations, the message flow through each action (for example, any defined switch action branches) is also shown. The date and time according to your user preferences are displayed.



6. Perform the following tasks in the **Activity Stream** section.

Click ...	To ...
	The first two options do not include the payloads of activities. • **Download JSON**: Download the activity stream data in JSON format. • **Download CSV**: Download the activity stream data in CSV format. • **Download Logs**: Download the instance activity stream logs. This log file includes the payload of activities.

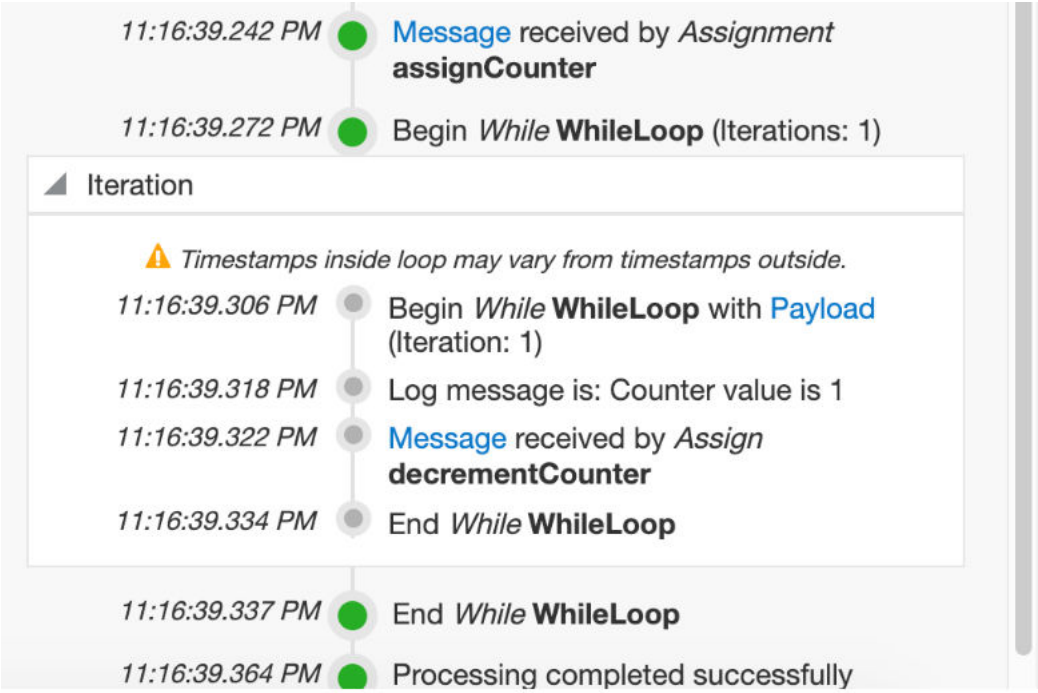
Click ...	To ...
	Display the activity stream on a full page. This link works as a toggle to expand and collapse the activity stream.
	Close the activity stream.

7. Click **Message** to view the payload details for the action.
8. Click

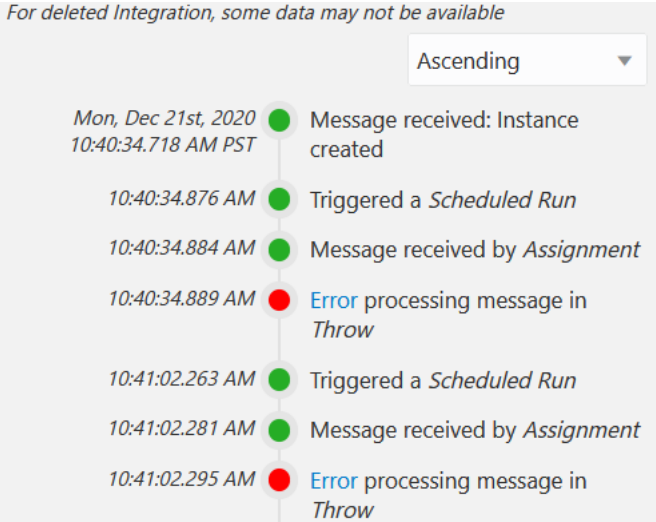
, then select an action to perform.

Click ...	To ...
Expand	Display the message on a full page. This link works as a toggle to expand and collapse the message.
Show Line Numbers	Number each line in the message.
Copy to Clipboard	Copy the message to the clipboard.
Download Payload	Download a text file of the payload.
Close	Close the message.

9. If your integration includes any looping actions (such as for-each, while, and others), expand **Iteration** to view flow execution inside the loop (available if you enabled tracing to include the payload when you activated your integration).



10. If the activity stream contains errors, click **Error** to show the error message.



11. Click

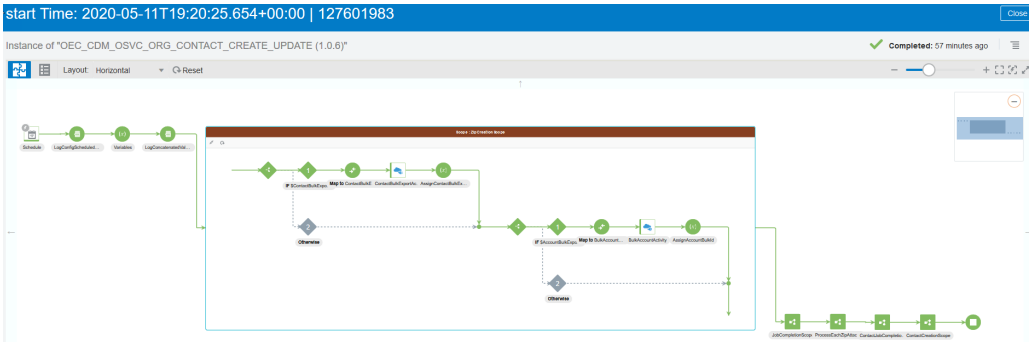
, then select an action to perform.

Click ...	To ...
Expand	Display the message on a full page. This link works as a toggle to expand and collapse the message.
Show Line Numbers	Number each line in the message.
Copy to Clipboard	Copy the message to the clipboard.
Download Payload	Download a text file of the payload.
Close	Close the message.

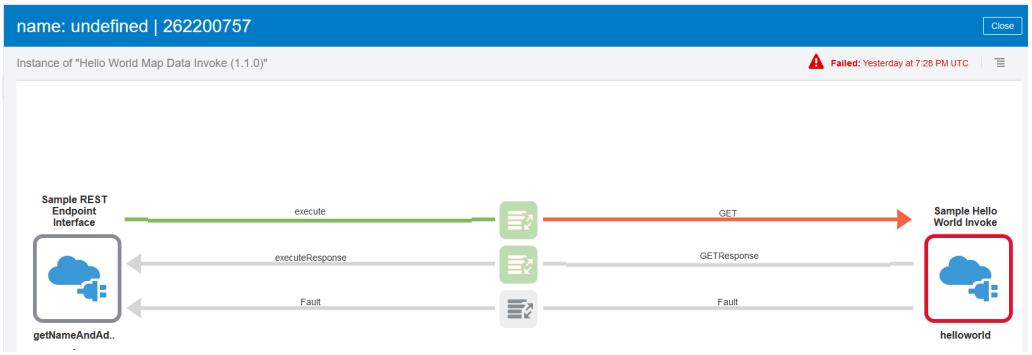
12. Return to the Track Instances page and click the business identifier to access a graphical display of the integration instance.

This page provides information about the business identifiers and values defined for the integration, the instance identifier of the integration, any error message, the activity stream, a button for discarding an error, and other information. The entire message flow is tracked through the integration. Successful message flows are indicated by green arrows. Any message flow errors that occur are identified by red arrows. For looping elements (such as for-each, while, and others), successful instances do not capture the flow inside the iterations. The loop-internal actions and colored lines are shown only if a failure occurs.

- For orchestrated integrations:



- For basic routing integrations:



The functionality available on this page is based on your type of integration style.

- View the Runtime Instance Flow of an Orchestrated Integration
- View the Runtime Instance Flow for a Basic Routing Integration

View the Runtime Instance Flow of an Orchestrated Integration

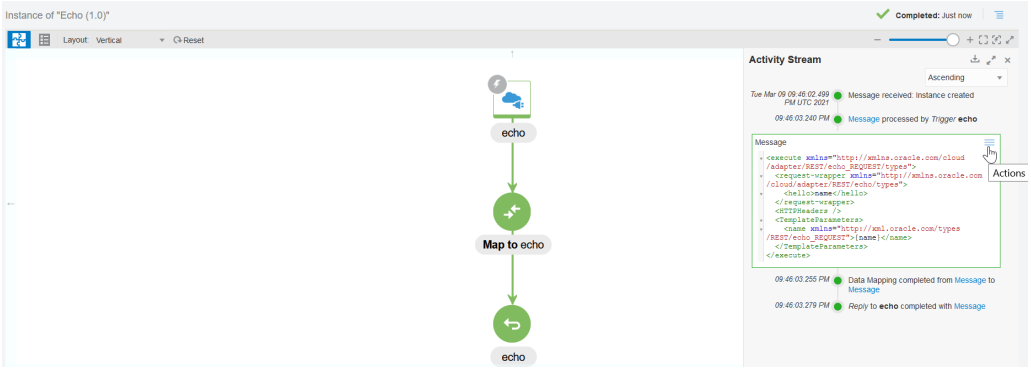
You can view a graphical display of the runtime instance flow of an orchestrated integration. The graphical display is shown when you click the business identifier for the instance on the Track Instances page.

Once on the page that shows the graphical display of the runtime instance, you can perform the following tasks:

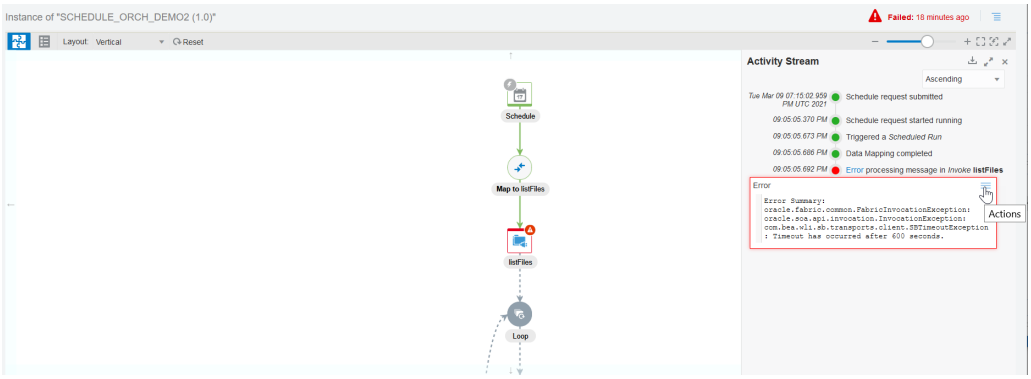
1. From the  menu in the upper right corner, select **View Activity Stream** to view the message flow. The activity stream shows details about the movement of the message through the integration, including where any failures occurred. For orchestrated integrations, the message flow through each action (for example, any defined switch action branches) is also shown. The date and time according to your user preferences are displayed.

   Perform similar activity stream tasks as described in Track Business Identifiers in Integrations During Runtime.

   - Successful instance:



- Failed instance:



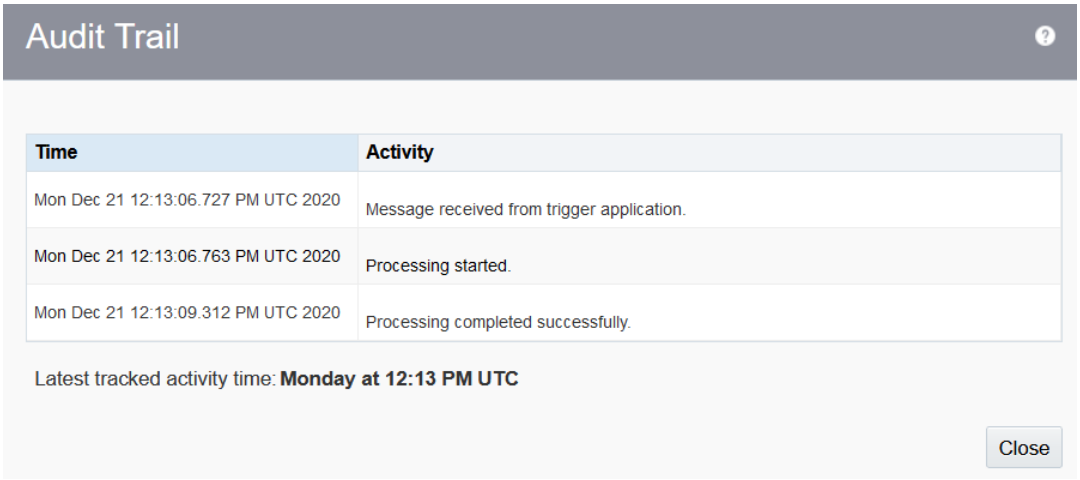
2. Click **Close** when complete.

View the Runtime Instance Flow for a Basic Routing Integration

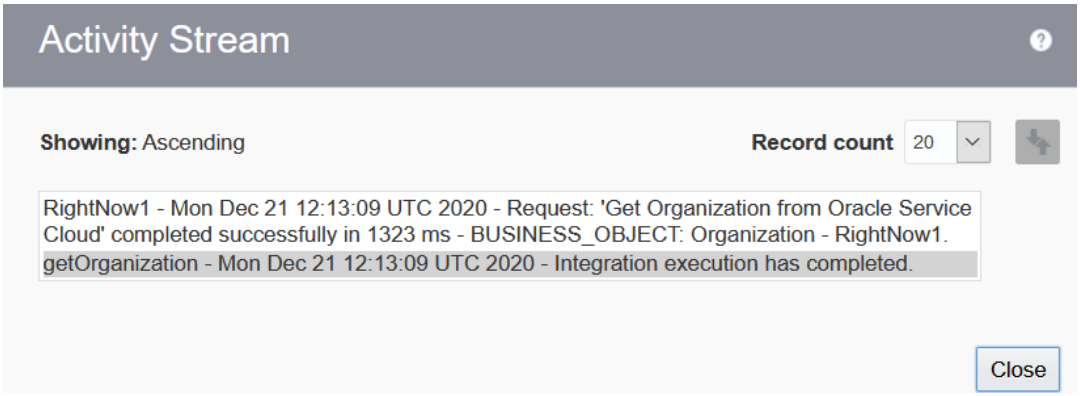
You can view a graphical display of the runtime instance flow of a basic routing integration. The graphical display is shown when you click the business identifier for the instance on the Track Instances page.

Once on the page that shows the graphical display of the runtime instance, you can perform the following tasks:

1. From the menu, select **View Audit Trail** to view the message flow. The audit trail shows details about the movement of the message through the integration, including where any failures occurred.



2. Select **View Activity Stream** to view the high level activity stream. If you enabled tracing to include the payload when you activated your integration, more specific payload details are provided.

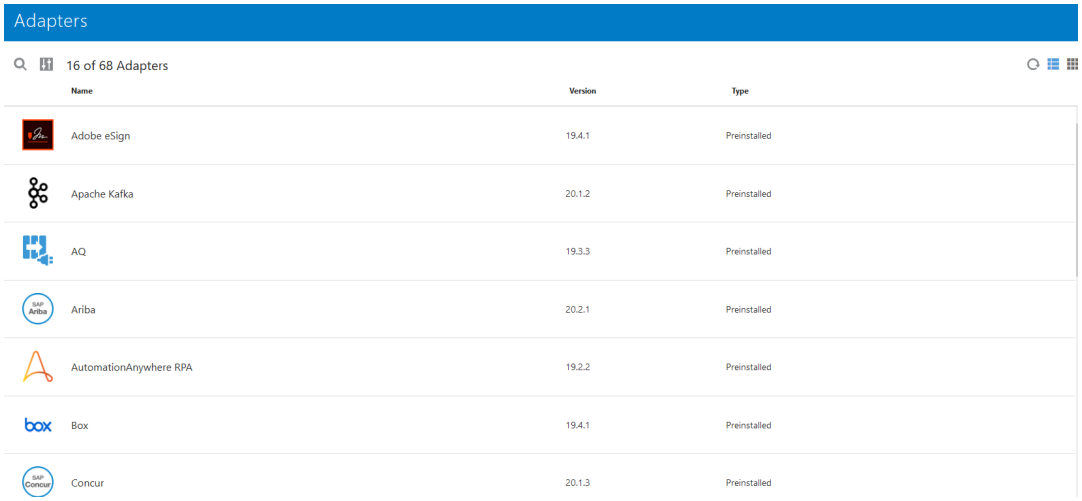


3. Select **Download Activity Stream** to download a ZIP file of the activity stream.
4. Click **Close** when complete.

View Preinstalled Adapters

You can view the adapters preinstalled with Oracle Integration.

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



2. Click

to filter by adapter type:
   - **Preinstalled**: Adapters automatically included with your instance of Oracle Integration.
   - **Private**: Custom adapters you created and uploaded to Oracle Integration.
   - **Marketplace**: Adapters included with packages imported from the Oracle Marketplace.
3. Click the adapter name or click  to display information about the adapter.
4. If you want to delete an adapter, select **Delete** from the

menu. You *cannot* delete adapters identified as **Preinstalled**.
5. If you want to create a connection with an adapter, select **Create Connection** from the

menu. The Create Connection dialog is displayed for that adapter.

Send Service Failure Alerts, System Status Reports, and Integration Error Reports by Notification Emails

You can notify users by email with reports about total messages received, total messages processed, successful messages, failed messages, and successful message rate; with service failure alerts (for example, when a service is down or an agent becomes unavailable); or with a

report every five minutes when an integration failure occurs. You can also remove email addresses from the suppression list.

> **Note:**
>
> - This is a system level setting. Therefore, user A cannot create one notification setup while user B creates another notification setup. User B can see the notification setup created initially by user A and can modify it.

Understand DKIM Authentication and SPF Record Details

When you configure notification emails to send service failure alerts, system status reports, and integration error reports, you can optionally specify Domain Keys Identified Mail (DKIM) authentication and Sender Policy Framework (SPF) record details.

Details and examples are provided to better understand DKIM and SPF. See the An Advanced Guide to OIC Notification via Emails blog and Configure Email Authentication Settings for SPF and DKIM in *Provisioning and Administering Oracle Integration Generation 2*.

Configure Notification Emails

You can send service failure alerts, system status reports, and integration error reports by notification emails. This section describes how to configure the notification emails.

1. In the left navigation pane, click **Home** > **Settings** > **Integrations** > **Notifications**.
2. Select when to send an email notification:
   - Send a critical health alert report every five minutes

     The report includes the following information:

     – POD name (also included in the subject of the email)

     – Time period

     – Total messages received, processed, succeeded, and failed

     – Percentage of successful messages

     – Error resubmittal count

     – Errors by integration instance and run including messages received, processed, succeeded, and failed, and an instance ID hyperlink that displays activity stream details, integration name, version number, and other information

     The email includes a link to the integration instance diagram that appears under the Track Instances page for a specific instance. The email also includes a link at the bottom to change the notification settings or completely stop receiving email notifications. Click **Oracle Integration** to access the Notifications page to adjust your settings.

- Send a detailed report every five minutes when an integration error is detected (for example, when a runtime or storage service is down or an agent becomes unavailable).
  - Reports are sent when one or more of the following conditions are detected:
    - A change in the status of any components.
    - The database quota exceeded the threshold.
    - The file system quota exceeded the threshold.
    - A quiesced or unquiesced action was performed.
    - One or more agent instances become unavailable.
    - Oracle Integration entered into read-only mode.
- Send an hourly detailed report
- Send a daily detailed report

3. In the **Recipients** field, enter the email address to which to send the reports and alerts. You can specify multiple email addresses, each separated by commas.
4. In the **Senders** section, click **+** in the lower right corner to add approved senders.
5. Specify the following details:

Element	Description
Email Address	An approved sender enables Oracle Integration to send email with a matching **From** address. An email is sent to each of the email addresses listed as approved senders. Upon approval, these email addresses can be used as the **From** address in a notification action of an orchestrated integration and/or system notification.
Approval Status	Indicates if the email address is approved. Green indicates the address is approved. Yellow indicates the address is not yet approved. Email address approval is based on your version of Oracle Integration. In Oracle Integration, a verification email is sent. You must click the verification link you receive in the email. Upon successful verification, status is changed to green. In Oracle Integration Generation 2, the email is automatically approved when you add the email ID.
SPF Status	Optionally configure the Sender Policy Framework (SPF) for the sender email addresses. You configure SPF support outside of Oracle Integration.
Confirm DKIM	Optionally confirm the Domain Keys Identified Mail (DKIM) for the sender email addresses. You configure DKIM support outside of Oracle Integration.
Use for System Notification	Select to use this email address for the system notifications you selected at the top of the page. However, only a single email address can be selected to send system notifications.

Recommendations are provided to ensure that you configure notification email correctly. See Troubleshoot Oracle Cloud Infrastructure Notification Email Configuration to Ensure Proper Delivery.

6. Click **Save**.
7. From the menu in the upper right corner, click **Send Now** to receive an immediate email about system status.
8. To stop receiving alerts and reports, clear the checkbox, then click **Yes** when prompted to confirm.
9. To clear the distribution list, selected alert/reports options, and specified sender email addresses, select **Reset All Notifications** from the menu in the upper right corner.

Remove Email Addresses from the Suppression List

You can remove email addresses from the suppression list.

> **Note:**
>
> This feature is only available in Oracle Integration Generation 2.

Adding users to a suppression list is a common task in email address management. Reasons for adding email addresses to the suppression list can include the following:

- An employee on an email distribution list left the company and their email address was deactivated.
- SPF and DKIM not being configured on the sender's address domain server (DNS) result in soft bounces and subsequently end up on the suppression list.

Oracle Integration provides a self-service feature that enables you to view and remove email addresses from the suppression list, as necessary. This feature eliminates the need to file a service request to remove addresses from the suppression list.

Suppressed email addresses can lead to errors in your integration. In the following example, the activity stream of the instance shows that the notification action failed because the email address used to send the notification is on the suppression list.

1. From the menu in the upper right corner, select **Email Suppression List**.

   The Email Suppression List drawer opens on the right side. The list of suppressed email addresses is displayed. Email addresses in the suppression list cannot be used to send or receive email notifications.

2. Select the email addresses to remove. You can also click

to search for addresses to remove.

> **Note:**
>
> The search functionality is case sensitive. For example, if the email address you are looking for is `joe.smith@mycompany.com`, and you enter `Joe.Smith@mycompany.com`, the email address is not returned in the search results.

3. Next to the email address to remove, click

.
This action removes this address from the suppression list.
4. Rerun the integration and check the activity stream again. The notification action should complete successfully this time.

Change the Delivery Time of Notification Emails

You can change the delivery time of notification emails.

The tracking metrics daily report executes by default at 00:00 UTC time, which can differ from your time zone. For example, the central time zone is 6 hours behind the UTC time zone. When the report runs at 00:00 UTC on November 14th, it is executing at 6:00pm central time on November 13th.

The report time is based on UTC time, which ensures a common reference point (particularly for the log files). This enables anyone in any time zone to identify the time of execution. This is important for global companies with operations in multiple countries/time zones.

As a workaround, set up notifications to run hourly and use a tool such an Outlook rule to automatically delete the 23 unwanted emails, keeping the one email with the desired tracking metrics.

See Troubleshoot Oracle Cloud Infrastructure Notification Email Configuration to Ensure Proper Delivery.

Set Logging Levels and Download Diagnostic Logs

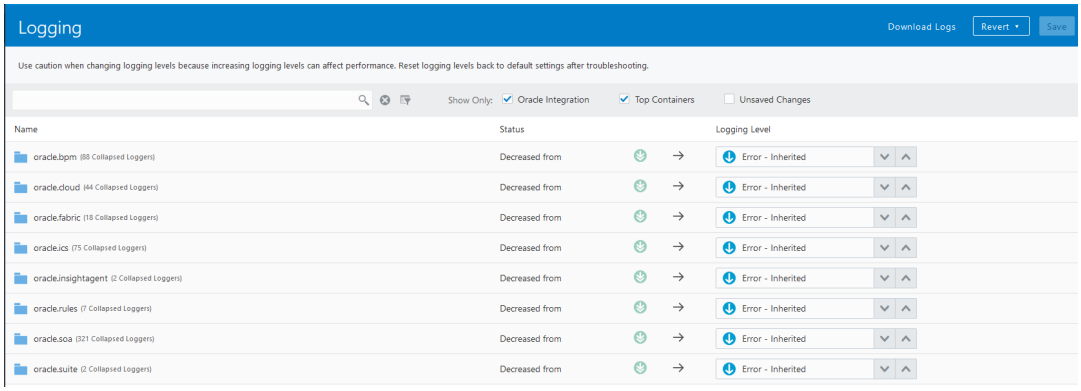
You can set logging levels and download diagnostic logs in Oracle Integration. You can also obtain the impacted POD name from the diagnostic logs. You cannot set logging levels or download logs in Oracle Integration Generation 2.

> **Note:**
>
> - Changing logging levels from their default values can fill up disk space and degrade system performance. Ensure that you reset the logger level to the default level when you have completed troubleshooting.
> - When you use the search facility, only the first ten results are displayed, even if there are more.

To set logging levels and download diagnostic logs.

1. In the left navigation pane, click **Home** > **Settings** > **Logging**. This selection is *not* available in Oracle Integration Generation 2.

   The loggers available for configuration are displayed. If the logger level is inherited from a parent logger, that information is displayed.



2. Change the logging level of any logger.
3. To revert to the previous setting before clicking **Save**, click **Revert**.
4. Click **Save**.

   A message is displayed at the top of the page:

   ```
   Logger levels updated successfully. Be sure to reset logger level to the
   default level when troubleshooting has been completed.
   ```

5. In the left navigation pane, click **Home** > **Monitoring** > **Integrations** > **Dashboard**.

> **Note:**
>
> You can also access the diagnostic logs directly from the Integrations page if your integration activation fails. Click the **Download diagnostic logs** link at the top of the page. See Activate an Integration.

6. From the **Logs** menu, select **Diagnostic Logs**. A message is displayed in the upper right corner.

   ```
   Downloading log file iscs-diagnostic-logs-number.zip has
   been started. Please check your browser's downloads list for progress and
   to access the file.
   ```

   You can get the POD name from inside the `environment.txt` file that is included in the ZIP file.

7. Unzip the file, and within the expanded directory structure, open *`AdminServer_number`*`-diagnostic.log`.

8. If the logger level is set to `NOTIFICATION`, this file also includes the WebLogic Server domain name (the POD name), which is written to the file every 30 minutes.

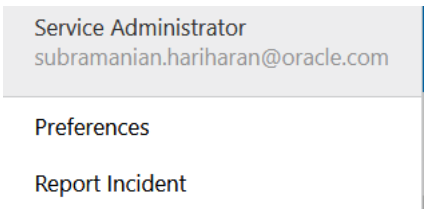
   An `environment.txt` file is part of the downloaded ZIP file. It also contains details regarding the Oracle Integration, including the version, mode, and POD name.

Report Incidents

You can report incidents for problematic issues that occur during design time (for example, being unable to open an integration, the failure of connection testing, or the failure of artifact regeneration).

> **Note:**
>
> The ability to report incidents has been deprecated.

1. In the left navigation pane, click **Home** > **Integrations** > **Integrations**.
2. From the *username* icon in the upper right corner of the page, select **Report Incident**.



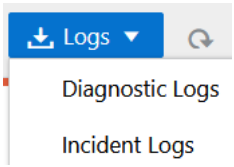
   The Report Incident dialog is displayed.

3. Enter a meaningful name and description of the incident to help others find and understand this incident.
4. Click **Create**.

An incident is created and a message is displayed at the top of the page. Remember the incident ID number. A **Download incident** link is also provided.

```
Incident report successfully created with ID: "number". You can download
it right now or later from the Monitoring dashboard.
```

5. Click **Home** > **Monitoring** > **Integrations** > **Dashboards**.
6. From the **Logs** menu, click **Incident Logs**. This menu is *not* available if you are using Oracle Integration Generation 2.



7. Enter the incident ID number, then click **Download**.
8. Save the ZIP file of incident logs to your local drive. The ZIP file includes a readme file with specific details, including the name and description you entered in the Report Incident dialog. If you open a service request with Oracle Support Services, share this information.

Purge and Retain Data in the Database

You can view the quiesced, unquiesced, and warning thresholds for your database and the percentage of the database that has been used. If you have the Administrator role, you can also set values for the quiesced threshold, unquiesced threshold, warning threshold, and retention times for successful and faulted instances. You can also perform both automatic and ad-hoc purges of the database.

> **Note:**
>
> Starting with the November 2020 release (20.4.2.0.0 (201009.0200.37961)), the page for purging and retaining data in the database has changed. See Set Data Retention for Runtime Instances in *Provisioning and Administering Oracle Integration Generation 2*.

Test Integration Instances

Oracle Integration provides the Oracle Asserter testing framework for recording tests of integration instances and replaying them to reproduce potential issues.

> **Note:**
>
> The Oracle Asserter testing framework has been deprecated.

You can simulate the behavior of both a trigger connection and the response from an invoke connection. You can also export integrations with recordings and import and test the integration in a different environment (for example, move it from a design environment to a test environment). You can also submit recordings to Oracle for analysis of errors.

Oracle Asserter enables you to perform the following:

- Create a test to run when fixing bugs in an integration. For example, an integration is deployed to production. Later, in the development instance, you change a mapping. If that breaks the integration because it now sends an incorrect message back to the client, Oracle Asserter can catch the regression.
- Test an integration in isolation of the dependent endpoints. For example, if an integration invokes certain endpoints and those endpoints are not accessible, you can bypass them and simulate the responses.
- Pass an integration (`.iar`) file and corresponding recording file to Oracle Support Services so they can reproduce a problem you are having without requiring access to remote endpoints such as SAP or Salesforce. With Oracle Asserter, the endpoints are simulated and not required to reproduce the issue.
- Ensure that the integration execution works across releases and different environments. You can automate the integration in your own repositories.

Restrictions

Note the following restrictions when using Oracle Asserter.

- Recording and playback of integration instances is only supported for trigger-based orchestrated integrations. Scheduled orchestrations are not supported.
- Asserter activations for integrations in which the connectivity agent is used in a trigger connection are not supported.
- Assertions can fail for integrations with invoke connections inside for-each actions.
- Playback works fine in reproducing fault instances. However, assertion evaluation times out and the proper results are not displayed in the Oracle Asserter Result dialog.
- There is a known issue with dynamic functions (for example, with the date function, the playback works for these integrations, but assertion evaluation fails. The results shown in the Oracle Asserter Result dialog are not correct.
- After the Oracle Asserter is enabled and any integration recording is played back for the first time, all the currently activated integrations stop serving the Swagger URL. To resolve this issue, deactivate and reactive all integrations again.
- If the total size of request and response payloads for all the invokes combined exceeds 25 MB, the recording is not created.

Record the Instance and Update Instance Details

When design is complete and the integration is ready for production, you can create an asserter test to check into your source repository.

This is because if you want to change the integration later, you can rely on the asserter test to catch regressions (for example, an assertion fails because the response you are sending to the client has changed due to a bug that was introduced in a mapping).

1. Create an orchestrated integration.
2. Go to the Integrations page and find the row for the integration.
   a. If the integration is not active, click the
   
   icon.

b. Select **Enable Asserter Recording** in the dialog that is displayed, then click **Activate**.

The integration is activated and the endpoint URL is displayed in the banner at the top of the page.

or

a. If the integration is already active, select **Enable Asserter Recording** from the 
menu.

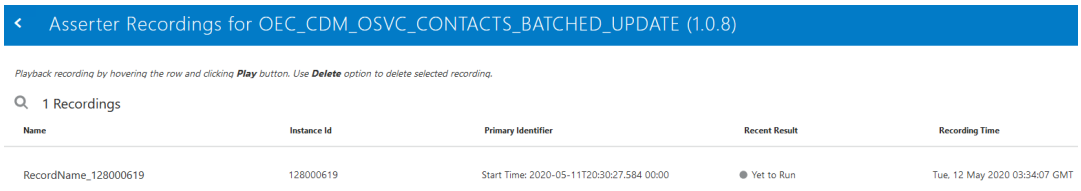
The following message is displayed:

```
Successfully enabled recording for integration integration_name (version).
```

3. Invoke the integration to create an instance.

If your integration includes multiple branches, create integration instances for each branch. For example, assume your integration includes a switch action that routes to one branch if the OrgID value passed is 7 or routes to an Otherwise branch if the OrgID value is not 7. Create two asserter recordings, one with each input value sent.

4. Select **Asserter Recordings** from the 
menu. The results are displayed. Each recording is identified by a unique value in the **Primary Identifier** column. You can create up to twelve recordings for a specific integration. When you create a twelfth recording, the first (oldest) recording you created gets deleted.



5. Go to the Track Instances page to see the instances.
6. Go to the row of the instance with the recording.
7. If you want to update the recording name and description, click 
.
8. Make your updates. You can only make these updates from the Track Instances page (and not from the Integrations page).
9. Click **Done**.

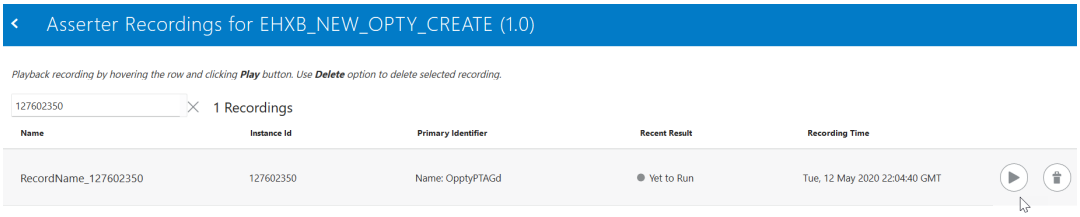
Play Back the Recorded Instance

You can play back the recorded instance using the same integration data.

1. Go to the Integrations page or Track Instances page.
   a. On the Integration page, find the integration that includes the recording.
   b. On the Track Instances page, find the integration instance that includes the recording.

2. Select the following:
   a. If on the Integrations page, select **Asserter Recordings** from the
   
   menu.
   b. If on the Track Instances page, select
   
   .
3. Select the row of the instance to replay, and click the

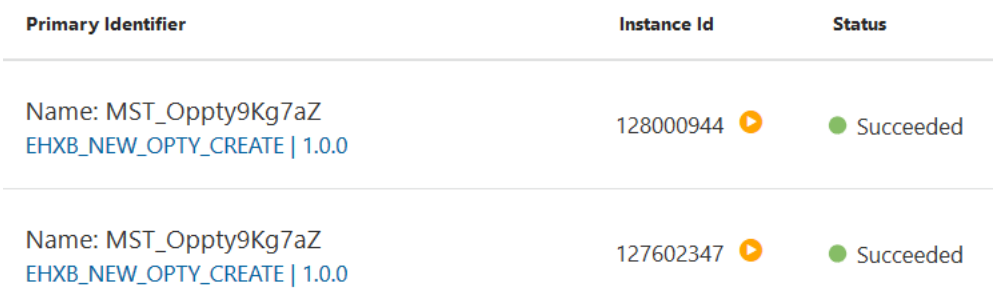
icon.



A message is displayed at the top of the page:

```
Successfully invoked playback for recording RecordName_number. Please
refer to Track Instances page to view the Asserter instance details.
```

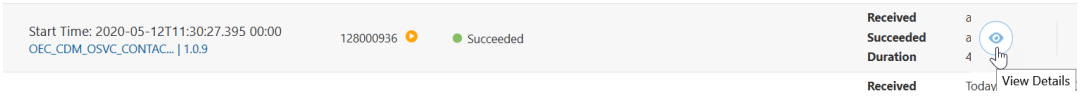
4. Return to the Track Instances page and note the icon shown next to the instance ID.



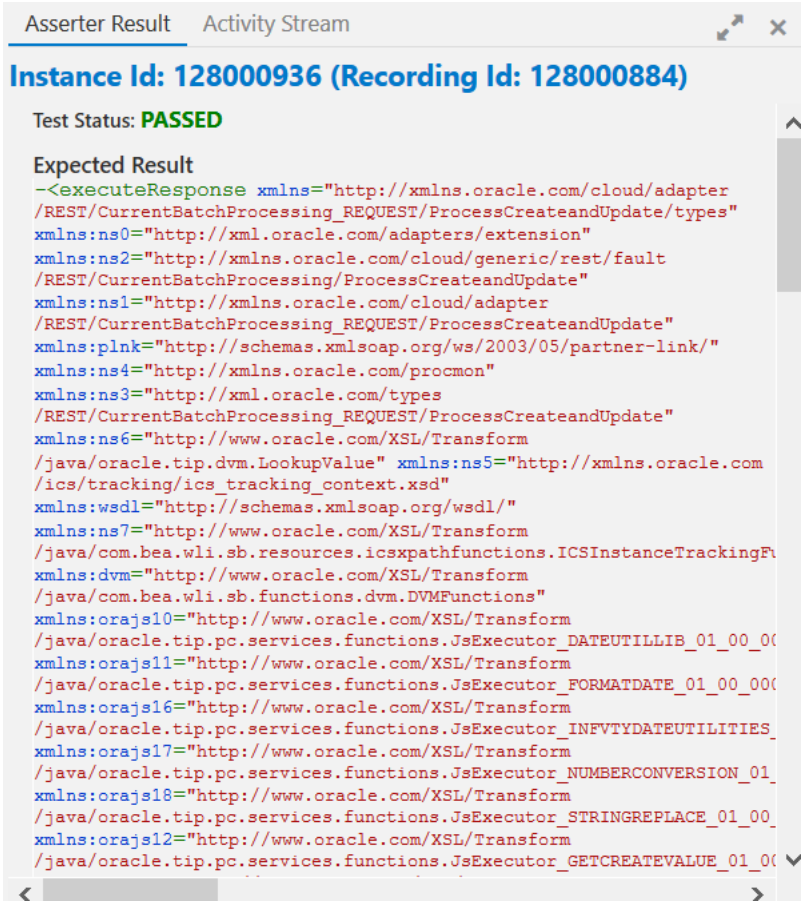
Monitor the Results with Assertions

You can monitor the results of integration instances that include assertions.

1. Select **Home** > **Monitoring** > **Integrations** > **Tracking** to track instance results and identify the asserter instance from the list of runs. To differentiate instances with asserter recordings from nonasserter recordings, the icon is displayed in the **Instance Id** column.
2. Select the **View Details** icon to view recording results.



The expected payload results are displayed at the top and the actual payload results are displayed below. This shows the input that was stored earlier as part of the recording that was compared against the actual response. If there is a match, the test passes. Otherwise, the test fails. This is an XML comparison and not a string comparison. Therefore, prefix differences are ignored.



> **Note:**
>
> Whenever a faulted instance is recorded (that is, the Track Instances page shows the instance as failed) and you perform a playback, the playback works fine in reproducing the fault. However, the assertion evaluation times out and you cannot see the proper results in the Oracle Asserter Result dialog.

Export and Import Integrations with Recordings

You can export and import integrations with recordings using the same export and import capabilities available with all integrations.

1. Select **Export** from the  menu.
2. Select the **Include Asserter Recordings** check box when prompted by the dialog.

> **Note:**
>
> When you export a package that includes or does not include recordings, the Export Package dialog shows the **Include Asserter Recordings** checkbox. If selected, the packages are exported with recordings. Otherwise, the recordings are not exported and cannot be imported.

3. Export the integration.
4. Go to the instance host on which to import the integration.
5. On the Integrations page, click **Import**.
6. Click **Browse** to find the integration (`iar`) file.
7. Select the **Include Asserter Recordings (if any)** check box, then click **Import**.

Delete a Recorded Instance

You can delete a recorded instance from the Integrations page or the Track Instances page.

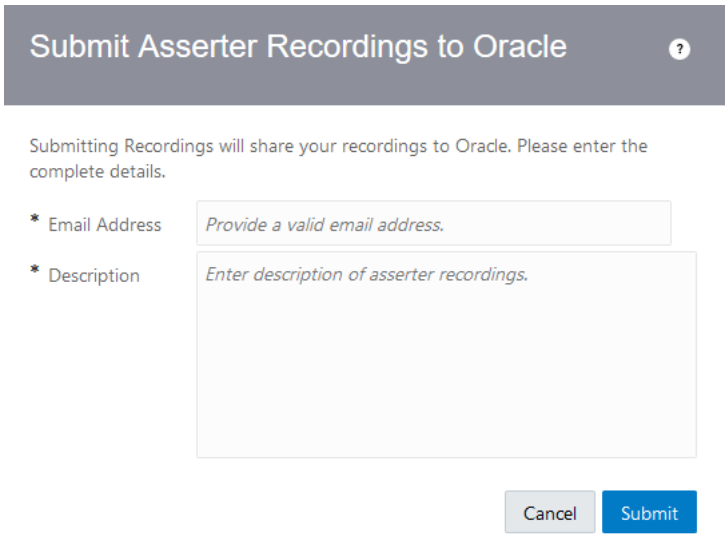
1. From the Integrations page:
   a. Go to the row of the integration that includes the recording to delete.
   b. Select **Asserter Recordings** from the menu.
   The Asserter Recordings for *integration_name* page is displayed.
   c. Find the recording to delete, and click
   .
2. From the Track Instances page:
   a. Go to the row of the integration instance that includes the recording to delete.
   b. Select
   .
   The Asserter Recordings for *integration_name* page is displayed.
   c. Find the recording to delete, and click
   .

Submit a Recording to Oracle Support Services

You can submit a recorded instance to Oracle. This action enables Oracle to play back the instance to reproduce the issue or bug.

1. On the Integrations page, find the integration that includes the recording to submit to Oracle.
2. Select **Submit Asserter Recordings** from the
   menu.

3. Enter your email address and a description of the recording. Oracle Support Services receives the recording and submits the results of their analysis to the specified email address.



13

Troubleshoot Oracle Integration

Learn about Oracle Integration troubleshooting tasks, including configuration recommendations to ensure proper email delivery, connectivity agent issues, integration runtime issues, integration design-time issues, scheduled integration issues, and integration activation issues.

> **Notes:**
>
> - There are overall service limits for Oracle Integration. A service limit is the quota or allowance set on a resource. See Service Limits in *Provisioning and Administering Oracle Integration Generation 2*.
> - Adapter troubleshooting information is provided in the individual adapter guides. See the Adapters page in the Oracle Help Center.
> - Cloud Customer Connect provides a community in which to ask questions, connect with experts, and share your thoughts and ideas about Oracle Integration.

Topics:

Troubleshoot Oracle Cloud Infrastructure Notification Email Configuration to Ensure Proper Delivery

Follow these recommendations to correctly configure and use the default `from` address and suppression list. These recommendations help you to avoid email delivery issues.

Default From Address

- Don't use `no-reply@oracle.com` as the `from` address.
- Don't use the `oracle` domain.
- Change the default `from` address from `no-reply@oracle.com` to `no-reply@mail.integration.`*`region`*`.ocp.oraclecloud.com`.
  The `region` attribute is provided by Oracle Integration.

- Change the `from` address in your integrations from `no-reply@oracle.com` to `no-reply@mail.integration.`*`region`*`.ocp.oraclecloud.com`.
  The `region` attribute is provided by Oracle Integration.

Suppression List

- Add `To` addresses to the suppression list for a number of reasons:
  - As of now, the `recipient` address when a hard bounce occurs (emails go undelivered for permanent reasons), when a soft bounce occurs (emails go undelivered for temporary reasons), and when a large number of emails are received are some of the reasons to add the `To` address to the suppression list.
- If DomainKeys Identified Mail (DKIM) and Sender Policy Framework (SPF) are not configured for the `from` address domain, the likelihood of having a bounce or messages being silently dropped by the receiving infrastructure is higher.
- You can remove email addresses from the suppression list. See Remove Email Addresses from the Suppression List.

Troubleshoot the Connectivity Agent

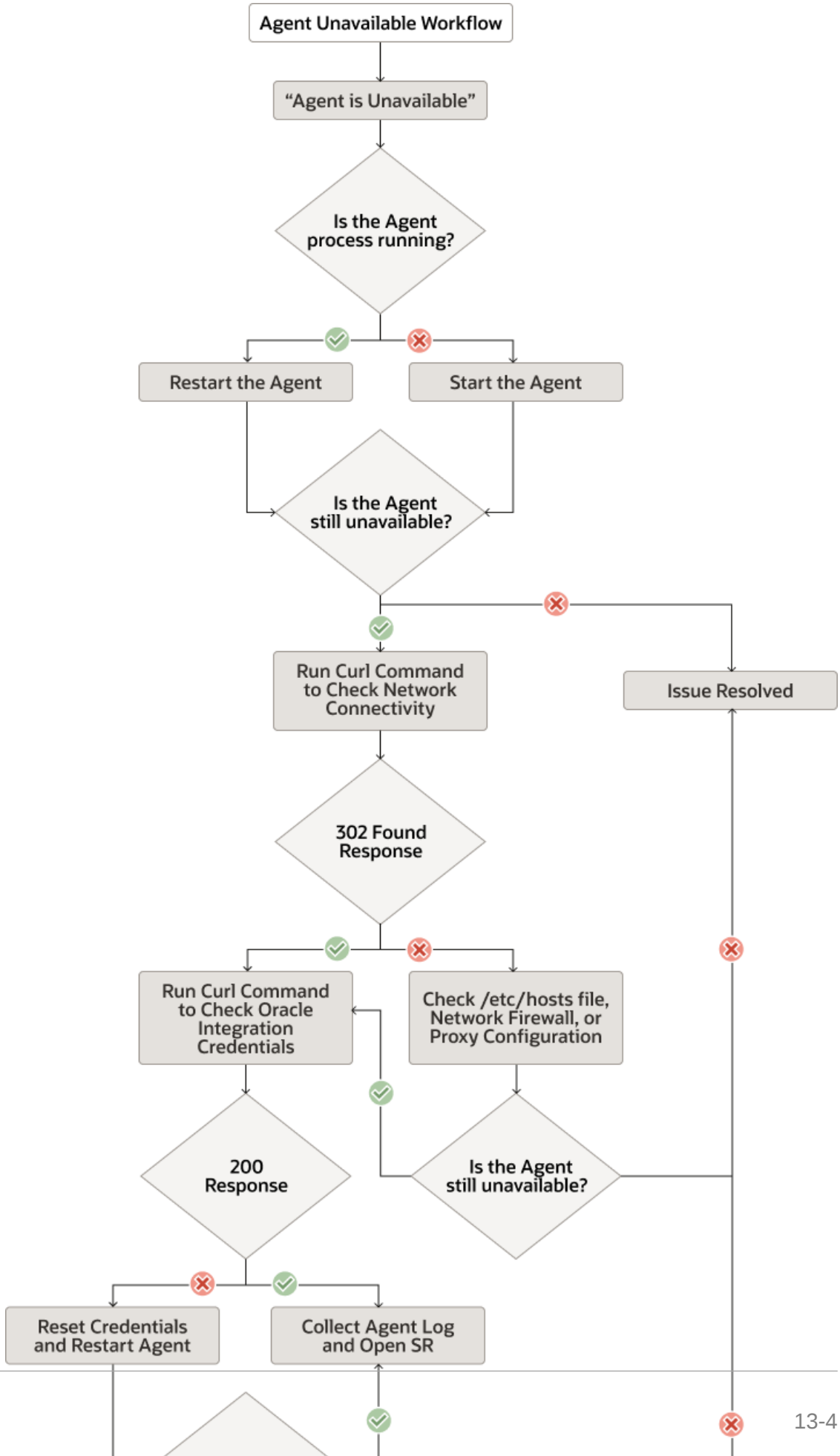
This section describes how to troubleshoot and resolve connectivity agent issues in Oracle Integration.

Topics:

- Integration Activation Error Due to Change in JDK Location
- Polling Flows on the Connectivity Agent Deactivated Due to Oracle Integration Being Quiesced or Unavailable
- Troubleshoot Network Connectivity Issues
- HTTP 401 Unauthorized Error Occurs During Connectivity Agent Installation

Troubleshoot Agent Unavailable

Connectivity agent status can be determined by selecting **Monitoring** > **Integrations** > **Agents** to access the Agents page. If the agent status is displayed as unavailable, follow this troubleshooting workflow.


Agent Unavailable Workflow
"Agent is Unavailable"
Is the Agent process running?
Restart the Agent
Start the Agent
Is the Agent still unavailable?
Run Curl Command to Check Network Connectivity
Issue Resolved
302 Found Response
Run Curl Command to Check Oracle Integration Credentials
Check /etc/hosts file, Network Firewall, or Proxy Configuration
200 Response
Is the Agent still unavailable?
Reset Credentials and Restart Agent
Collect Agent Log and Open SR

Number	Step	Details

1	**Check the agent process is running**	Check whether the agent is running. Run one of these commands from the host where the agent is installed.
		If you don't see `connectivityagent.jar` as a listed process, the agent is not running. For example: `19054 connectivityagent.jar`
		Linux:
		`ps -ef | grep connectivityagent.jar`
		Windows:
		`jps -l`
		If the agent process is not running, start it:
		Linux:
		Start the connectivity agent as a background process:
		`nohup java -jar connectivityagent.jar &`
		Windows:
		`java -jar connectivityagent.jar`
		Note:
		Starting the agent as a Windows service is not supported.
		If you try to start the agent but it won't start, continue to check network connectivity.
		If the agent process is running, restart it:
		Follow these steps to restart the agent.

Number	Step	Details
2	**Check network connectivity**	**Check that the agent host can connect to the Oracle Integration instance:** On the host where the agent is installed, run this command, replacing with your own path: `curl https://myoic-tenancy-ia.integration.ocp.oraclecloud.com/ic/home` **If the response is 302 Found**, the agent is able to resolve the Oracle Integration instance host. You have network connectivity. Continue to check user credentials. **If the response is not 302 Found**, there may be network issues. Check the `/etc/hosts` file: 1. Check for the correct IP address and host name for your Oracle Integration instance. 2. Troubleshoot with your network administrator to check connectivity to the instance is not blocked by a firewall or proxy. Here's agent troubleshooting information.

Number	Step	Details

3	**Check user credentials**	Check that the user account used to run the agent has a valid user name and password:
		1. Sign in to Oracle Integration with the user account and password that are used to run the agent. This confirms that the user account and password are valid.
		2. Check that the user credentials used by the agent are valid. On the host where the agent is installed, run this command using the credentials you used to sign in to Oracle Integration and replace OIC_HOSTNAME with your host name:
		`curl -v -k -X GET -u user_name:password https://OIC_HOSTNAME/icsapis/v2/environment`
		If the response to checking user credentials is 200, the user name and password used by the agent are valid. There may be something else wrong. Check the diagnostic log for errors:
		`AGENT_INSTALL_LOCATION/agenthome/logs/agent-diagnostic0.log`
		• If you see in the log `401 Unauthorized`, or `Agent startup failed: Unable to get agent status`, or `Agent Instance status check failed. Response received`, there's an issue with the user configured in the agent `InstallerProfile.cfg` file, the user account is locked, or the password may have expired. Follow the steps in *If the response to checking user credentials is not 200*.
		If the response to checking user credentials is not 200, the credentials the agent is using aren't valid. The user might be locked, the password might have expired, or the `InstallerProfile.cfg` may be corrupt.
		Reset user credentials and start the agent:
		1. Check whether the user is locked, and if so, unlock the user and set a new password. Follow these steps to unlock the user account for the agent.
		2. Sign in to Oracle Integration using the new credentials to check that they work.
		3. Update the user name and password in the `InstallerProfile.cfg` file, on the host where the agent is installed.
		`oic_USER=Oracle_Integration_username`
		`oic_PASSWORD=Oracle_Integration_password`
		4. Start the agent.
		Linux:

Number	Step	Details

		Start the connectivity agent as a background process:
		`nohup java -jar connectivityagent.jar &`
		Windows:
		`java -jar connectivityagent.jar`
		Note:
		Starting the agent as a Windows service is not supported.

4	**Log a Service Request**	If you've gone through the troubleshooting steps, can't find anything in the diagnostic log, and the agent status is still unavailable, create a service request with Oracle Support. Log in, select the **Service Requests** tab, and click **Create Technical SR**. Include the diagnostic log in the service request.

Unable to Start Connectivity Agent Because Agent Status Cannot Be Obtained

Connectivity agent startup can fail with the following error:

```
java -jar connectivityagent.jar &



[Logger: oracle.cloud.cpi.agent.util.CpiUtils] [SRC Class:
oracle.cloud.cpi.agent.util.CpiUtils;
 Method: checkIfAlreadyRunning] Agent Instance status check failed. Response
received -

[Logger: oracle.cloud.cpi.agent.ProxyCpiAgentLoader] [SRC Class:
oracle.cloud.cpi.agent.ProxyCpiAgentLoader;
Method: initialize] Agent Startup Failed java.lang.RuntimeException: Unable
to get agent status  :
at
oracle.cloud.cpi.agent.util.CpiUtils.checkIfAlreadyRunning(CpiUtils.java:1058)
at
oracle.cloud.cpi.agent.ProxyCpiAgentLoader.initialize(ProxyCpiAgentLoader.java
:114)
at
oracle.cloud.cpi.agent.ProxyCpiAgentLoader.runAgent(ProxyCpiAgentLoader.java:5
3)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at
sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.j
```

```
ava:43)
at java.lang.reflect.Method.invoke(Method.java:498)
at oracle.cloud.cpi.agent.CpiAgentLoader.startAgent(CpiAgentLoader.java:27)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at
sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.j
ava:43)
at java.lang.reflect.Method.invoke(Method.java:498)
at
oracle.cloud.cpi.agent.CpiBootstrapLoader.startAgent(CpiBootstrapLoader.java:1
83)
at
oracle.cloud.cpi.agent.AgentStartCommandTask.run(CpiBootstrapLoader.java:427)
at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
at java.util.concurrent.FutureTask.run(FutureTask.java:266)
at
java.util.concurrent.ScheduledThreadPoolExecutor$ScheduledFutureTask.access$20
1(ScheduledThreadPoolExecutor.java:180)
at
java.util.concurrent.ScheduledThreadPoolExecutor$ScheduledFutureTask.run(Sched
uledThreadPoolExecutor.java:293)
at
java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149
)
at
java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624
)
at java.lang.Thread.run(Thread.java:748)

 [Logger: oracle.cloud.cpi.agent.AgentStartCommandTask] [SRC Class:
oracle.cloud.cpi.agent.AgentStartCommandTask; Method: run]
 Error during agent install/startup or while re-starting agent after
upgrade:  java.lang.reflect.InvocationTargetException
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at
sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.j
ava:43)
at java.lang.reflect.Method.invoke(Method.java:498)
at
oracle.cloud.cpi.agent.CpiBootstrapLoader.startAgent(CpiBootstrapLoader.java:1
83)
at
oracle.cloud.cpi.agent.AgentStartCommandTask.run(CpiBootstrapLoader.java:427)
at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
at java.util.concurrent.FutureTask.run(FutureTask.java:266)
at
java.util.concurrent.ScheduledThreadPoolExecutor$ScheduledFutureTask.access$20
1(ScheduledThreadPoolExecutor.java:180)
at
java.util.concurrent.ScheduledThreadPoolExecutor$ScheduledFutureTask.run(Sched
uledThreadPoolExecutor.java:293)
at
```

```
java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149
)
at
java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624
)
at java.lang.Thread.run(Thread.java:748)
Caused by: java.lang.reflect.InvocationTargetException
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at
sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.j
ava:43)
at java.lang.reflect.Method.invoke(Method.java:498)
at oracle.cloud.cpi.agent.CpiAgentLoader.startAgent(CpiAgentLoader.java:27)
... 13 more
Caused by: java.lang.RuntimeException: Agent Startup Failed - Unable to get
agent status
```

This error occurs because the connectivity agent cannot query the agent status from the Oracle Integration server. Common causes includes connectivity issues and incorrect/expired credentials in the connectivity agent configuration file.

1. Execute the following command using the credentials specified in the `InstallerProfile.cfg` file during connectivity agent installation:

   ```
   curl -v -k -X GET -u uid:pwd  https://Oracle_Integration_URL:443/
   icsapis/v2/environment
   ```

   If the `curl` command returns a `401 Unauthorized` error, that means the server is reachable and has returned a response indicating that invalid credentials were sent to the server.

2. Update the credentials in the `InstallerProfile.cfg` file:

   a. Back up `InstallerProfile.cfg`.

   b. Change the user name and password fields to values used with your earlier REST test. Ensure that you can log in to Oracle Integration using these credentials.

   ```
   oic_USER=Oracle_Integration_username
   oic_PASSWORD=Oracle_Integration_password
   ```

3. Save the file.

4. Restart the connectivity agent.
   Upon a successful restart, the connectivity agent updates the entries with encrypted values.

Connectivity Agent Startup Fails With a "java.lang.RuntimeException: Agent Startup Failed - Unable to get agent status" Error

Connectivity agent startup can fail with the following error:

```
[2021-06-19T10:06:06.567Z] [SEVERE] [ThreadID: 1] [Logger:
oracle.cloud.cpi.agent.CpiBootstrapLoader]
[SRC Class: oracle.cloud.cpi.agent.CpiBootstrapLoader; Method: main] Error
```

```
encountered !!! java.lang.RuntimeException:
 java.lang.reflect.InvocationTargetException
at
oracle.cloud.cpi.agent.AgentStartCommandTask.run(CpiBootstrapLoader.java:452)
at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
at java.util.concurrent.FutureTask.run(FutureTask.java:266)
at
java.util.concurrent.ScheduledThreadPoolExecutor$ScheduledFutureTask.access$20
1(ScheduledThreadPoolExecutor.java:180)
at
java.util.concurrent.ScheduledThreadPoolExecutor$ScheduledFutureTask.run(Sched
uledThreadPoolExecutor.java:293)
at
java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149
)
at
java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624
)
at java.lang.Thread.run(Thread.java:748)
Caused by: java.lang.reflect.InvocationTargetException
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at
sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.j
ava:43)
at java.lang.reflect.Method.invoke(Method.java:498)
at
oracle.cloud.cpi.agent.CpiBootstrapLoader.startAgent(CpiBootstrapLoader.java:1
92)
at
oracle.cloud.cpi.agent.AgentStartCommandTask.run(CpiBootstrapLoader.java:436)
...
...
Caused by: java.lang.reflect.InvocationTargetException
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at
sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.j
ava:43)
at java.lang.reflect.Method.invoke(Method.java:498)
at oracle.cloud.cpi.agent.CpiAgentLoader.startAgent(CpiAgentLoader.java:27)
...
...
Caused by: java.lang.RuntimeException: Agent Startup Failed - Unable to get
agent status :
at
oracle.cloud.cpi.agent.ProxyCpiAgentLoader.initialize(ProxyCpiAgentLoader.java
:151)
at
oracle.cloud.cpi.agent.ProxyCpiAgentLoader.runAgent(ProxyCpiAgentLoader.java:5
3)
...
...
```

The issue occurs because the user account configured for the connectivity agent is locked or the password has expired.

1. Enter the following `curl` command and see whether you receive a status `200` response. Ensure that you use the exact same username and password as used for the connectivity agent configuration.

   ```
   curl -v -k -X GET -u uid:pwd https://Oracle_Integration_URL:443/icsapis/v2/
   environment
   ```

   If the above `curl` command response is not `200`, there is an issue with the user configured in the `InstallerProfile.cfg` file.
2. Check the user account in the Oracle Cloud Infrastructure Console and whether the user is locked or the password has expired. Try logging in to Oracle Integration with the exact username and password credentials and ensure that the user can get into Oracle Integration.
3. Once the user can get into Oracle Integration, update the `oic_USER` and `oic_PASSWORD` parameters in the `InstallerProfile.cfg` file with the correct values and restart the agent. This should address the issue.

Connectivity Agent Installation Error if /etc/hosts File is Not Configured

During connectivity agent installation, if the agent installer cannot find details about the host name/virtual machine (VM) on which the agent is being installed, you may receive the following error:

```
Caused by: java.net.UnknownHostException: : Unknown
name or service
```

Resolve this error by adding an entry for the IP address and host name in the `/etc/hosts` file of the host name/VM.

Restrictions on Using the Stage File Action with the Connectivity Agent

Be aware of several restrictions when using some operations of the stage file action with the connectivity agent. See Restrictions on Using Stage File Action Operations with the File/Attachment Features of the Connectivity Agent.

Connectivity Agent Installation Fails When Run as a Federated User

Connectivity agent installation fails when run as a federated user. If you rely on user federation, you must create a nonfederated user to use specifically for installing the connectivity agent. This user enables the connectivity agent to communicate with Oracle Integration.

> **Note:**
>
> If you ever need to restart the connectivity agent, you must ensure that the username/password credentials for this user are still valid.

Java Memory Errors

The agent Java Virtual Machine can stop running when allocating memory with the following seemingly out-of-memory error:

```
There is insufficient memory for the Java Runtime Environment to continue.
-----------------------------
Java HotSpot(TM) 64-Bit Server VM warning: INFO:
os::commit_memory(0x00007f6847afd000, 12288, 0) failed; error='Cannot
allocate memory' (errno=12)
#
# There is insufficient memory for the Java Runtime Environment to continue.
# Native memory allocation (mmap) failed to map 12288 bytes for committing
reserved memory.
# An error report file with more information is saved as:
```

However, this issue is unrelated to the out-of-memory error that is sometimes observed when the Java heap is not large enough.

This error occurs when Java requests more memory from the operating system, which does not have any:

```
# There is insufficient memory for the Java Runtime Environment to continue.
# Native memory allocation (mmap) failed to map
```

This may be related to a physical machine or server or virtual machine resources on the physical machine/virtual machine/server on which the agent is executing. As an example, when the agent is running on the same virtual machine as the database, and the database may be consuming most of the resources. Oracle recommends that you set up the agent on a separate compute.

Authorization Error When Restarting the Agent

If you manually stop the agent and attempt to restart it, and receive an authorization error, ensure that the user name and password used to start the agent in the `InstallerProfile.cfg` file are correct. This error can occur if the password for this user name expired and was changed in the Oracle Cloud Infrastructure Console by the administrator, but was not updated in the `InstallerProfile.cfg` file. This task is *only* required if you manually stop and restart the agent. This task is not required for agent upgrades, which occur automatically and do not use these credentials.

Add or Change the Non-Proxy Host Configuration After Installation in the CpiAgent.properties File

If you need to add, change, or bypass the non-proxy host configuration *after* agent installation, do *not* edit the `InstallerConfing.cfg` file. Proxy host changes made to that file after agent installation do not take effect. Instead, update the host with the `proxy_nonProxyHosts` parameter in the *`Agent_Installation_Location`*`/agenthome/agent/config/CpiAgent.properties` file for your changes to take effect. After editing this file, restart the agent.

Class Loading Conflict When Sharing the JDK Instance with Another Product

When using the connectivity agent, the following error occurs because your JDK instance is shared with another product for which JAR files have been added in the JDK's endorsed directory. This results in a class loading conflict with the agent:

```
ClassCastException: com.sun.xml.messaging.saaj.soap.ver1_1.Message1_1Impl
cannot be cast to oracle.j2ee.ws.saaj.soap.MessageImpl
```

Ensure that your connectivity agent is running with a JDK installation that is not modified because of use with other products.

Connectivity Agent Log File Location

The connectivity agent `agent-diagnostic0.log` file is available under `agenthome/logs`.

Error When Using the Connectivity Agent in an Oracle Integration Classic (User-Managed) Environment

If you are using Oracle Integration Classic (user-managed) and agent interactions fail with the following error:

```
java.sql.SQLException: ORA-03146: Invalid buffer length for TTC field
```

You must apply the following patch to the Oracle Database Cloud Service instance used with your Oracle Integration instance.

1. Go to http://support.oracle.com and obtain patch 26482376.
2. Apply the patch to the Oracle Database Cloud Service instance.
3. Run the following command against the database. (Note that running this command helps even without applying the patch.)

   ```
   alter system set events '24921 trace name context forever, level=105989
   ```

Agent Behavior in a Decommissioned Instance or HTTP 404/401 Error Response Codes

The following code in the logs indicates that agent run time message processing has halted. This occurs if an HTTP 404/401 error code is received by the agent for a continuous period of 24 hours. The decommission of an Oracle Integration instance also triggers this behavior. When the conditions leading to this error have been resolved, the agent must be restarted manually. See Restart the Agent.

```
[2018-10-13T04:30:13.501Z] [SEVERE] [ThreadID: 18] [Logger:
oracle.cloud.cpi.agent.transport.AQRuntimeConsumer] [SRC Class:
oracle.cloud.cpi.agent.transport.AQRuntimeConsumer; Method: run] Terminate
flag activated. Signalling termination of agent runtime poller thread with Id

CPI_RUNTIME_REQ_RES_CHANNEL_worker_0.
```

Verify Endpoint Accessibility when Agent is Installed with a Proxy Host

When the agent is installed with a proxy host, carefully check that the endpoint to access through the agent is reachable through the proxy host. If it is not reachable through the proxy

host, you must configure the on-premises endpoint host in the `proxy_NON_PROXY_HOSTS` parameter of the *`Agent_Installation_Location`*`/agenthome/agent/config / CpiAgent.properties` file.

Unlock the Agent Group

When an agent group is in edit mode and the browser crashes, the agent group becomes locked, which prevents it from being edited. This results in the following error:

```
ICS-10507: The agent group cannot be updated because it is locked.
```

To unlock the agent group:

1. Log in again as the same user who was editing the agent group when the browser crashed, then log out. This action unlocks the agent group.

or

1. Wait 30 minutes for the lock to expire after the timeout starts.

Failure to Send a Response Due to a java.net.SocketException: Connection Reset Error

For connectivity agent installations running on an Oracle Integration Classic VM and connecting to Oracle Integration (running on Oracle Cloud Infrastructure), the design time and runtime operation involving the connectivity agent sometime fails with a `java.net.SocketException: Connection reset` error.

This can occur because of a Maximum Transmission Unit (MTU) mismatch.

Here is the complete error:

```
[2019-01-03T16:35:12.670Z] [SEVERE] [ThreadID: 50] [Logger:
oracle.cloud.cpi.agent.transport.CpiAgentAQTransporter] [SRC Class:
oracle.cloud.cpi.agent.transport.CpiAgentAQTransporter; Method:
sendOneWayPacket] Exception while sending response back to
ICSjava.net.SocketException: Connection reset
com.sun.jersey.api.client.ClientHandlerException: java.net.SocketException:
Connection reset
at
com.sun.jersey.client.urlconnection.URLConnectionClientHandler.handle(URLConne

ctionClientHandler.java:155)
at com.sun.jersey.api.client.Client.handle(Client.java:652)
at com.sun.jersey.api.client.WebResource.handle(WebResource.java:682)
at com.sun.jersey.api.client.WebResource.access$200(WebResource.java:74)
```

To make this connection work, set the MTU of the Oracle Integration Classic VM (where the agent is installed) to `1500` from the current value of `8900`.

Perform the following steps:

1. Run `ifconfig -a` as the root user (sudo) and note down the network interface.
2. Run the following command as the root user (sudo) for the network interface (assuming the network interface is `eth0`):

   ```
   ifconfig eth0 mtu 1500
   ```

Anytime the agent VM is restarted (note that it is not an agent restart, but the host where the agent is installed), the changes must be done for the network interface before restarting the agent.

Agent Installation on Linux Fails When Using an Installer Copied Using winscp

During connectivity agent installation in Linux environments, the installation sometimes fails with the following error:

```
On premise agent is throwing the following error: java.lang.RuntimeException:

Agent Startup Failed - java.lang.IllegalArgumentException: URI is not
absolute
```

Installation failure occurs for the following reason:

1. The agent is to be installed in a Linux environment.
2. The agent installer ZIP file is downloaded in a Windows environment and transferred to a Linux environment for installation using a Windows tool called `winscp`.
3. Even when the binary option is enabled in the `winscp` tool, the installer ZIP used to run and install the agent fails with the above error.

As a workaround, perform the following steps:

1. Download the agent installer to a Linux environment directly and do not transfer it from a Windows environment.
2. If the Oracle Integration user interface is accessible from a Linux environment, use the download install option provided on the Agents page.
3. If the Oracle Integration user interface instance is not accessible, use the following REST command to download the installer to a Linux environment:

   ```
   curl -k -v -X GET -u OIC_user:OIC_password
   -H 'Content-Type:application/json'
   'https://OIC_URL:443/icsapis/v1/agent/binaries/connectivity'
   -o download_location/oic_connectivity_agent.zip
   ```

Agent Performance Tuning

Only modify the `agentWorkerThreads` property in the *`Agent_Installation_Location`*`/agenthome/agent/config/CpiAgent.properties` file when Oracle Integration has been scaled out to handle additional loads. In that case, the agent can be tuned to handle additional loads by changing the `agentWorkerThreads` property value. The maximum value that can be assigned to each agent is `10`.

Integration Activation Error Due to Change in JDK Location

If integration activation fails with the following error, this is likely the result of the agent installation using a JDK whose location has been changed (for example, removed). This can occur if the agent was installed and running with a version of the JDK whose location was removed and a newer version was installed in a different location. If the JDK installed with the

agent is removed, ensure that you restart the agent with the newer version (and location) of the JDK.

```
Caused by: java.lang.Error: Circular loading of installed providers
detected at
java.nio.file.spi.FileSystemProvider.installedProviders(FileSystemProvider.jav

a:161)
at java.nio.file.FileSystems.newFileSystem(FileSystems.java:324)
at java.nio.file.FileSystems.newFileSystem(FileSystems.java:276)
at
oracle.cloud.cpi.agent.store.StoreUtils.extractZipBundle(StoreUtils.java:49)
at
oracle.cloud.cpi.agent.store.FileAgentStore.addActivation(FileAgentStore.java:
40)
at
oracle.cloud.cpi.agent.ActivationCpiCommand.doRequest(ActivationCpiCommand.jav

a:80)
```

Polling Flows on the Connectivity Agent Deactivated Due to Oracle Integration Being Quiesced or Unavailable

If Oracle Integration is quiesced or unavailable, any polling flows on the connectivity agent (database, JMS, file, and so on) used as a trigger are deactivated. The flows are then reactivated when Oracle Integration is unquiesced or becomes available again for message processing. However, if Oracle Integration remains unavailable for more than five minutes, the polling flows are deactivated on the agent side. The connectivity agent must be restarted for the trigger endpoints to be reactivated and resume polling of EIS endpoints.

Troubleshoot Network Connectivity Issues

Ensure that network connectivity is working correctly when persistent connectivity failures are encountered with an Oracle Integration instance.

> **Note:**
>
> This does not apply to any intermittent failures because the connectivity agent is resilient to temporary conditions and recovers when the situation is resolved.

- Run the following command:

  ```
  nslookup hostname
  ```

- Run the following command for a period of five minutes to also capture any transient failures:

  ```
  while true;
  do
  curl https://Oracle_Integration_hostname/ic/home >> file.txt 2>&1;
  sleep 1;
  echo "trying again";
  done;
  ```

HTTP 401 Unauthorized Error Occurs During Connectivity Agent Installation

If connectivity agent installation or restart fails with an `HTTP 401 Unauthorized` error, it implies the username/password used to bootstrap the connectivity agent are incorrect. Ensure that the username/password credentials being used to install/restart the connectivity agent are valid. You can use `postman` or `curl` to access the following API and ensure that it succeeds with an `HTTP 200` response for the username/password specified. That eliminates any bad username/password combination you may currently be using.

- Option 1 - Use `postman` or `curl`:

  ```
  curl -k -X GET -u user:password https://Oracle_Integration_Host/icsapis/v2/environment
  ```

  If the above API succeeds with an `HTTP 200` response, use the same username and password to also bootstrap the connectivity agent.
- Option 1 - Use a Chrome browser or any other browser to run the following:

  ```
  https://Oracle_Integration_Host/icsapis/v2/environment
  ```

Troubleshoot Integration Runtime

Learn about troubleshooting integration runtime in Oracle Integration.

Topics:

Using Basic Authentication to Call Services When Oracle Identity Cloud Service is Configured With Multi-Factor Authentication Causes a 401-Unauthorized Error

If Oracle Identity Cloud Service is configured with multi-factor authentication (for example, user name / password + multi-factor authentication) and you use basic authentication to invoke Oracle Integration services, you receive a `401-Unauthorized` error. None of the adapters or the connectivity agent in Oracle Integration support multi-factor authentication.

As an alternative:

- Disable multi-factor authentication from the account with which you are using basic authentication.
- Use OAuth instead.

See Managing Multi-Factor Authentication.

Perform Remedial Actions When the Payload Exceeds the Size Limit

When the message payload in an integration exceeds the size limit, the message is rejected and an error message is displayed. The error message describes remedial actions to perform.

Issue	Error Message	Remedial Action to Perform
Payload size limit exceeded	• When REST, File, SOAP, FTP, and SAP Adapter connections that support payload sizes of up to 50 MB exceed the limit: `CASDK-0059: Unable to process the response received from the client application. The response received by the OIC integration flow is rejected. The response received has been rejected since the content received of length 5723727000 bytes exceeds the maximum allowed threshold of 52428800 bytes (50MB).`  • When all other adapter connections that support payload sizes of up to 10 MB exceed the limit:  `CASDK-0059: Unable to process the response received from the client application. The response received by the OIC integration flow is rejected. The response received has been rejected since the content received of length 1723727000 bytes exceeds the maximum allowed threshold of 10485760 bytes (10MB).`	The preferred way to send large data sets is by using attachments. Check whether the target application exposes a SOAP API to return data sent as an attachment. If yes, the integration can be designed to configure an invoke connection using the SOAP Adapter to accept an MTOM attachment. The attachment can be an archive containing an XML/comma-separated value (CSV) document.
Attachment size limit exceeded	`CASDK-0059: Unable to process the response received from the client application. The response received by the OIC integration flow is rejected. The response received has been rejected since the content received of length bytes exceeds the maximum allowed threshold of 1073741824 bytes (1GB).`	Check whether the attachment can be broken and sent through multiple APIs. For example, if the attachment is a ZIP archive containing a large CSV file and unstructured documents, have one API that returns the CSV file and another API that retrieves the unstructured documents.

For additional details about 50 MB payload support, limits, and best practices, see Service Limits in *Provisioning and Administering Oracle Integration Generation 2*.

Remove Unprintable Control Characters from the Incoming Payload

Oracle Integration does not support certain non-printable control characters in the following range.

```
['\u0000', '\u0001', '\u0002', '\u0003', '\u0004', '\u0005', '\u0006',
'\u0007', '\u0008', '\u000b', '\u000c', '\u000e', '\u000f', '\u0010', '
\u0011', '\u0012', '\u0013', '\u0014', '\u0015', '\u0016', '\u0017', '\
u0018', '\u0019']
```

The presence of these control characters in the incoming payload results in the following error during processing:

```
{
  "type" : "10.4.1",
  "title" : "exception occurred while translating the request from native to
xml.",
  "detail" : "oracle.cloud.connector.api.CloudInvocationException:
ORABPEL-15235\n\nTranslation Failure.\nFailed to translate JSON to XML.
org.codehaus.jackson.JsonParseException: Illegal unquoted character
((CTRL-CHAR, code 1)): has to be escaped using backslash to be included in
string value\n at [Source: java.io.BufferedReader@.#; line: 2, column:
22]\nThe incoming data does not conform to the NXSD schema. Please correct
the problem.\n",
  "o:errorCode" : "TRANSLATION-ERROR-00",
  "o:errorDetails" : [ {
    "type" : "NA",
    "instance" : "NA",
    "title" : "NA",
    "o:errorPath" : "NA",
    "o:errorCode" : "NA"
  } ]
}
```

If you receive this error, contact Oracle Support for information on how to preprocess incoming JSON payloads and filter unprintable control characters.

Notification Action Returns a <from> value is invalid Error

If the notification action in an orchestrated integration validates successfully, but the following error occurs during integration runtime, the value specified in the **Parameter Value** of the notification action may not be correct.

```
:
summary=<summary>The <from> value is invalid.
Xpath expression associated with <from> in copy assign activity is invalid.
There is an user error that results in missing element value(s) in the xpath
query.
Please review the payload and modeling to ensure that all elements defined in
the <from> xpath query have valid non-null values .
</summary>
```

```
:
,query=<query>fn:current-dateTime()</query>
```

The error can be confirmed by viewing errors for the integration instance on the Track Instances page or the response of a REST testing tool such as `Postman`.

Verify that the value specified in the **Parameter Value** of the notification action is correct. For example, this value may require conversion to a string:

```
string(fn:current-dateTime())
```

Synchronous Integrations Occasionally Fail with a 504 Gateway Timeout Error

Synchronous integrations can fail with the following error.

```
Bad response: 504 Gateway Time-out from url
https://Instance_URL/Integration_Name/Version_Number/
```

This error occurs because synchronous integrations (integrations that return a response payload) are limited to 300 seconds and return a timeout error if they exceed that limit.

Optimize your integration and/or the endpoint you invoke to complete instance execution in under 300 seconds.

See Service Limits in *Provisioning and Administering Oracle Integration Generation 2*.

Invalid <From> Value When Using a Fault Handler in an Integration

An integration configured with a fault handler can fail with the following error indicating that the `<from>` value is invalid.

```
ICSWC-0001) :
Error message is - The <from> value is invalid.
Xpath expression associated with <from> in copy assign activity is invalid.
There is a user error that results in missing element value(s) in the xpath
query.
Please review the payload and modeling to ensure that all elements defined in
the <from> xpath query have valid non-null values.
```

The fault format indicates that the endpoint configuration predates the fault handling infrastructure. Therefore, this fault cannot be handled inside a fault handler associated with a scope or default handler. This occurs when you import a pre-17.4.5 integration from Oracle Integration Cloud Service into Oracle Integration.

```
<soap-env:Fault xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
<faultcode
xmlns:faultsrc="http://xmlns.oracle.com/cloud/adapter/REST/
GetLocation_REQUEST">faultsrc:APIInvocationError</faultcode>
<faultstring>ICS runtime execution error</faultstring>
<detail>
<ICSfaultVar/>
```

```
<reason><genericRestFault><errorCode>400</errorCode><er
rorPath&gt;&lt;![CDATA[GET
https://myinsance.com/maxrest_b1dk/rest/os/ICSLOCATION?
_lid=icsuser&_lpwd=M%24lee019&LOCATION=101104&STATUS=OPERATING returned a
response status of 400 Bad
Request]]></errorPath><instance><![CDATA[Error 400:
BMXAA7901E - You cannot log in at this time. Contact the system
administrator. ]]></instance></genericRestFault&gt;
:Application Error</reason>
<operation>execute</operation>
<request_payload/>
</detail>
</soap-env:Fault>
```

Solution

Do not modify the integration to include fault handling. If you need to modify, edit the adapter endpoint configuration. After editing the adapter endpoint configuration, the integration is updated to throw a typed fault conforming to the `APIInvocationError` format mentioned below. This enables the fault to be handled inside fault handlers.

```
<soap-env:Fault xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
<faultcode
xmlns:faultsrc="http://xmlns.oracle.com/cloud/adapter/REST/
InvokeMonitoring_REQUEST">faultsrc:APIInvocationError</faultcode>
<faultstring>ICS runtime execution error</faultstring>
<detail>
<ICSfaultVar>
<nstrgdfl:APIInvocationError
xmlns:nstrgdfl="http://xmlns.oracle.com/cloud/generic/rest/fault/REST/
InvokeMonitoring">
<nstrgdfl:type/>
<nstrgdfl:title/>
<nstrgdfl:detail/>
<nstrgdfl:errorCode/>
<nstrgdfl:errorDetails>
<nstrgdfl:type>http://www.w3.org/Protocols/rfc2616/rfc2616-
sec10.html#sec10.4.1</nstrgdfl:type>
<nstrgdfl:title>Bad Request</nstrgdfl:title>
<nstrgdfl:errorCode>400</nstrgdfl:errorCode>
<nstrgdfl:errorPath>&lt;![CDATA[POST
http://myinstance.com:7003/ic/api/integration/v1/monitoring/errors/(id)/
resubmit returned a response status of 400 Bad
Request]]></nstrgdfl:errorPath>
<nstrgdfl:instance>&lt;![CDATA[{"status":"HTTP 400 Bad
Request","title":"Invalid ID format. Please provide only
numbers.","type":"https://www.w3.org/Protocols/rfc2616/rfc2616-
sec10.html#sec10.4.1"}.A 400 Bad Request Error indicates that the target
service is unable
(or refuses) to process the request sent by the client (Oracle Integration
Cloud), due to an issue that is perceived by the server to be a client
problem. You can trace the cURL representation of the request sent to the
target service from the Oracle Integration Cloud server logs. Try invoking
the target service using cURL. It may also be that one of the intermediaries
(proxy, LBR) could be returning this code. ]]&gt;</nstrgdfl:instance>
```

```
</nstrgdfl:errorDetails>
</nstrgdfl:APIInvocationError>
</ICSfaultVar>
<reason>Fault Details : <nstrgdfl:APIInvocationError
xmlns:nstrgdfl="http://xmlns.oracle.com/cloud/generic/rest/fault/REST/
InvokeMonitoring"><nstrgdfl:type/><nstrgdfl:title/><nstrgdfl:detail/
><nstrgdfl:errorCode/><nstrgdfl:errorDetails&gt;&lt;nstrgdfl:type&
gt;http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10.4.1</
nstrgdfl:type><nstrgdfl:title>Bad
Request</nstrgdfl:title><nstrgdfl:errorCode>400</nstrgdfl:error
Code><nstrgdfl:errorPath&gt;&amp;lt;![CDATA[POST
http://myinstance.com:7003/ic/api/integration/v1/monitoring/errors/(id)/
resubmit returned a response status of 400 Bad
Request]]></nstrgdfl:errorPath&gt;&lt;nstrgdfl:instance&gt;&amp;lt;!
[CDATA[{"status":"HTTP 400 Bad Request","title":"Invalid ID format. Please
provide only
numbers.","type":"https://www.w3.org/Protocols/rfc2616/rfc2616-
sec10.html#sec10.4.1"}.A 400 Bad Request Error indicates that the target
service is unable
(or refuses) to process the request sent by the client (Oracle Integration
Cloud), due to an issue that is perceived by the server to be a client
problem. You can trace the cURL representation of the request sent to the
target service from the Oracle Integration Cloud server logs. Try invoking
the target service using cURL. It may also be that one of the intermediaries
(proxy, LBR) could be returning this code.
]]></nstrgdfl:instance></nstrgdfl:errorDetails&gt;&lt;/nstrgdf
l:APIInvocationError> :Application Error</reason>
<operation>execute</operation>
<request_payload/>
</detail>
</soap-env:Fault>
```

Activity Stream Logs Deleted

The activity stream logs in Oracle Integration can be automatically deleted under certain scenarios. These scenarios are described below, along with links to documentation that describes how to capture the activity stream logs in the Oracle Cloud Infrastructure Console, enable tracing with the payload and view the activity stream logs on the Track Instance page, and download the activity stream logs with the REST APIs.

- Automatic Rotation of the Activity Stream Log
- Capture the Activity Stream in the Oracle Cloud Infrastructure Console
- Enable Tracing with the Payload and View the Activity Stream on the Track Instances Page
- Download the Activity Stream Logs with the REST APIs

Automatic Rotation of the Activity Stream Log

Activity stream logs in Oracle Integration can be automatically deleted under the following scenarios:

- You have too many integrations with tracing and payload logging enabled.
- You have too many instances of an integration.

The activity stream log size is 100 MB. If either of these scenarios occurs, the file is rotated and you do not see the activity stream. Oracle does not recommend enabling tracing and payload logging in production environments.

Instead, design an integration that can push the payload and other information such as instance ID, user, integration name, code, version, and so on to the storage location of your choice and call this integration from critical integrations through a local invoke connection to persist the log information.

The activity stream payload data is derived from the `ics-flow.log` files that are created on the integration instance. These files are rotated after they reach a size of 10 MB, and only 10 of these files are retained. There is no option to reconfigure the file size or file rotation settings. Depending on the number of transactions (instance IDs generated) and payload sizes, the flow log files can be rotated frequently. This limits the time window during which the activity stream can show payload and trace data. This is the expected behavior when selecting the **Enable tracing** and **Include payload** options during integration activation.

There is no guarantee that the activity stream payload data is persisted or available for a certain amount of time. The availability of activity stream data depends on your integrations. The **Enable tracing** and **Include payload** options are intended for design-time and troubleshooting use. Use these options in a production only when absolutely necessary. As a best practice, set business identifiers to track individual fields of the payload for monitoring and debugging purposes. These tracking fields are displayed for each integration instance in the Track Instances page and are retained for the entire instance retention period.

To reduce the size of integration logs (and thereby increase payload data availability in the activity stream), you can do the following:

- Only activate your integrations using the **Enable tracing** and **Include payload** options when absolutely required (for debugging purposes).
- Do not set the **Include payload** option at activation for integrations that handle very large payloads or a high number of payloads (for example, through a looping mechanism).

Capture the Activity Stream in the Oracle Cloud Infrastructure Console

Documentation is available that describes how to capture the activity stream in the Oracle Cloud Infrastructure Console:

- Capture the Activity Stream of Integrations in Oracle Cloud Infrastructure Console in *Provisioning and Administering Oracle Integration Generation 2*

Enable Tracing with the Payload and View the Activity Stream on the Track Instances Page

Documentation is available that describes how to enable tracing with the payload and view the activity stream for orchestrated integrations on the Track Instances page:

- See Enable or Disable Tracing on Active Integrations
- See Activate an Integration
- See View the Runtime Instance Flow of an Orchestrated Integration

Download the Activity Stream Logs with the REST APIs

Documentation is available that describes how to download activity stream logs in Oracle Integration or with the REST APIs:

- View the Activity Stream from the Dashboard Page
- See Retrieve Activity Stream Messages

- See Download a Log File (by specifying the `icsflowlog` parameter)

> **Note:**
>
> If you decide to use absolute values (10 MB, instead of a certain size), downloading the file downloads the latest 10 MB file only. The rotated files are not available for download.

Message Dequeuing Takes Three to Four Seconds in Asynchronous Integrations

You may observe that Oracle Integration sometimes takes three to four seconds to go from `Message Received:Instance created and enqueued for processing` to `message dequeued and processed by Trigger` *`trigger_name`*. This increases the end-to-end processing time of a particular instance.

This is the expected behavior with asynchronous requests. Asynchronous requests are processed using a queue. The first two entries in the flow logs show the enqueue and dequeue times. Therefore, a few seconds lag is expected.

403 Error When Accessing an API Protected Using OAuth

OAuth allows Oracle Integration to access a user's resources on their behalf. REST and other REST-based adapters (such as Google Calendar, Microsoft Email, and others) often access APIs that are protected using OAuth. If these calls result in a `403` error (forbidden), it usually implies a lack of permissions to access the API.

You need to carefully examine your OAuth configuration and work with the target API owner/administrator to resolve this issue. There may be several conditions leading to this, but the exact cause is best described by the authorization server. Some probable causes of `403` errors are as follows:

- The access token was procured for a scope that does not cover the API being accessed.
- During the phase to provide consent, the user that provided the consent is not the resource owner or owner of the API being accessed.
- The access token expired or is not valid to access the resource any longer.
- The authorization server revoked access privileges to a particular resource.

Troubleshoot Certificate Import Failures

The certificate that you receive from a Certificate Authority (CA) such as Verisign, Entrust, or others can fail during import into Oracle Integration with the following error in the log file:

```
java.security.cert.CertificateException: Unable to initialize,
java.io.IOException: extra data given to DerValue constructor
```

To resolve this issue, either:

- Obtain a fixed certificate from the CA.

- Convert the certificate to a Distinguished Encoding Rules (DER) binary certificate using tools such as `openssl`. For example:

```
openssl x509 -outform der -in sfdc-client.crt -out sfdc-client.der
```

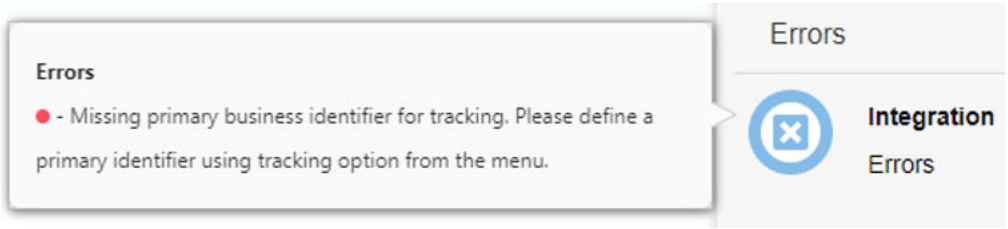
Troubleshoot Integration Design Time

Learn about troubleshooting integration design time in Oracle Integration.

Topics:

Missing a Primary Business Identifier in an Integration

You must configure at least one business identifier tracking field to make your integration ready for activation. This is known as the primary business identifier. Business identifiers enable you to track payload fields in messages during runtime. If you have not yet configured at least one business identifier, the following error message is displayed when you hover your cursor over the **Integration Errors** icon in the design canvas of your integration.



Configure at least one business identifier to remove this error message. See Assign Business Identifiers.

Unable to Call a replaceAll JavaScript Function

If you attempt to call a `replaceAll` JavaScript function in Oracle Integration, you receive the following error:

```
ode=<code>JavaScript execution error</code>\n,detail=<detail>Review
JavaScript error(s) and fix them:\nerror: null@3 -> TypeError: Cannot find
function replaceAll in object
```

The `replaceAll` is not supported by the JavaScript engine in Oracle Integration.

Use one of the following alternatives to achieve `replaceAll` functionality:

- Use the `replace` function with the global option, as follows:

```
function myFunction(Input) {
  var res = Input.replace(/<\/>/g,'');
 return res;
}
```

- Use the `split` and `join` functions instead of `replaceAll`.

Conflict with Using Variables of String Data Type and Applying Mathematical Comparator Expressions

All variables created in Oracle Integration are currently only string data types. Using these variables directly and applying mathematical comparator expressions results in unusual behavior.

For example, in an assign action, two variables, `var1` and `var2`, are created with values of `1` and `10`, respectively. In a while action, if you then set the condition as `$var1 < $var2`, the loop terminates after two iterations. The correct usage is to set the condition by wrapping the variables with a `number` XPath function: `number($var1) < number($var2)`. This loop iterates 10 times before terminating.

Reduce Network Calls that Slow Down Processes Caused By Writing Record By Record

Writing record by record using the **Append to Existing File** option on the Operations page of the Adapter Endpoint Configuration Wizard of the FTP Adapter creates too many network calls and eventually slows down the process. Instead, use a stage file action and write the records to a stage directory. Once all records are written, use the **List File** option in the stage file action and the FTP Adapter **Write File** option to transfer the file to an FTP location. This approach reduces processing time and prevents too many calls to the FTP server.

Invoke Subprocess Integrations Using a Single REST API Connection

Assume you follow a modular design approach in which you create separate integrations (think of them as subprocesses) to perform specific tasks. For example:

- File - Staging
- Translation
- Stage - Core
- Trigger Job Processes

Each of these subprocesses is created as a WSDL connector. However, having to maintain multiple connectors complicates the situation. Ideally, creating a single REST API and triggering subprocesses using API calls would be easier to maintain. Note the following details:

- You can invoke subprocesses using a single REST API connection.
- You can use a “schedule now” API to trigger the required integration. However, this is only possible if the orchestrated integration was created as a scheduled integration. In addition, the “schedule now” API runs the specified integration immediately and this type of integration cannot be configured to take input parameters.

As an alternative, you can create a nonscheduled integration with a REST connection used as the trigger that can be configured to take input parameters. That integration can be invoked as a REST API itself, without the need for the “schedule now” API to invoke it. The URL for the “schedule now” API in the v2 version of the REST API is:

```
http://host:port/icsapis/v2/integrations/SAYHELLO%7c01.00.0000/schedule/jobs
```

That is, the URL contains `CODE|VERSION` to specify which integration to run.

Note that a SOAP connector is the only option available that can call subprocesses.

Trigger Connection Cannot Receive an Empty Array in JSON if the Corresponding JSON Received Has an Empty JSON

An adapter does not receive any element from the mapper if the element is an empty array. To get the element from the mapper in case of an empty array, change the XSL file manually.

1. Export the integration.
2. Manually change the mappings in the XSL file to ensure that the default elements are generated in the target even when the source elements do not exist or are empty. For example:

   Before editing the XSL file:

   ```
   <xsl:for-each
   select="$Cheers_API/nsmpr0:executeResponse/nsmpr1:response-wrapper/nsmpr1:
   employeeEarnedDetails/nsmpr1:earnedBadges" xml:id="id_24">
   <nstrgdfl:earnedBadges xml:id="id_23">
   <xsl:value-of select="." xml:id="id_25" />
   </nstrgdfl:earnedBadges>
   </xsl:for-each>
   ```

   After editing the XSL file:

   ```
   <nstrgdfl:earnedBadges xml:id="id_23">
   <xsl:for-each
   select="$Cheers_API/nsmpr0:executeResponse/nsmpr1:response-wrapper/nsmpr1:
   employeeEarnedDetails/nsmpr1:earnedBadges" xml:id="id_24">
   <xsl:value-of select="." xml:id="id_25" />
   </xsl:for-each>
   </nstrgdfl:earnedBadges>
   ```

3. Import the integration.

Troubleshoot Scheduled Integrations

Learn about troubleshooting scheduled integrations in Oracle Integration.

Topics:

Scheduled Integrations Are Not Executing on Time

Scheduled integrations may be queued and not execute on time as scheduled for the following reasons.

- There is a running instance of the same scheduled integration. No new instance of the same integration is executed until the running instance completes.
- The throttling limit for scheduled integrations has been reached. No new scheduled integrations are executed until the number of scheduled integrations being executed falls below the throttling limit.

> **Note:**
>
> The throttling limit is automatically adjusted based on available system resources.

When a Scheduled Integration Instance Gets Terminated

An instance of a scheduled integration can be terminated by a user or Oracle Integration for the following reasons.

- A user cancels a scheduled flow.
- A user call reactivates a scheduled integration, which causes the currently-queued scheduled instance to be terminated.
- A scheduled flow execution exceeds six hours, which causes Oracle Integration to terminate the scheduled instance.

Reasons for Oracle Integration to Terminate a Scheduled Instance

There are several reasons for Oracle Integration to terminate a scheduled instances.

- To prevent an integration execution from occupying the limited resources allocated for scheduled integrations for a long time, which can block other scheduled integrations.
- To prevent a stuck integration (for unexpected reasons) from blocking subsequent runs of the same integration.
  A scheduled integration is considered stuck under the following circumstances:
  - For a scheduled integration without parallel processing (that is, no parallel for-each, reading of multiple files, or stage file action read file in segments operation), the total execution time exceeds six hours.
  - For a scheduled integration with parallel processing (that is, parallel for-each, reading of multiple files, or stage file action read file in segments operation), the execution time of a batch or pre/post execution of a parallel execution exceeds six hours.

Troubleshoot Integration Activations

Learn about troubleshooting integration activations in Oracle Integration.

Topics:

- Delay in Message Processing of First Instance after Activation

Troubleshoot Integration Activations

If your integration fails to activate, review the following failure issues and corrective actions.

Activation Failure Causes	Corrective Actions
WSDL issues: • Duplicate schema issues in the WSDL • WSDL/schema validation failures	1. Check the WSDL for duplicate schemas. 2. Check that the WSDL is valid. 3. If necessary, correct the WSDL in the application. 4. Reconfigure the integration by regenerating the endpoints. 5. Retry the activation.
Deployment time out issues.	Retry the activation because the server may be slow to react.

Database issues:
• Reached the size limit:
`java.sql.SQLException: ORA-01654: unable to extend index`
• Database is down.
• Connection pool issues:
`Could not create pool connection for datasource 'SOALocalTxDataSource'. The DBMS driver exception was: Listener refused the connection with the following error: ORA-12516, TNS:listener could not find available handler with matching protocol stack`
	1. Check that the database is running and has enough space.
	2. If not, the database administrator must increase the database size limit or connection pool size.
	3. Retry the activation.

A connection is down or was modified.	1. Check if the service/application is accessible. If it is down, retry the activation later. 2. If the connection is invalid, update the connection configuration. 3. Reconfigure the integration by regenerating the endpoints. 4. Retry the activation.
Managed server unavailable.	1. The administrator must restart the server. 2. Retry the activation.

Activation Failure Causes	Corrective Actions
If an integration contains too many actions (for example, it has 25 or more stage write actions), activation failure can occur. The diagnostic logs show a `51 code too large` error: `RegularFileObject[/u01/data/domains/iscs /servers/integration_server1/dc/ 338c94 f1-d3ee-45b9-bbf8-43c9203df851/SCA-INF/ bpel/ StageCurrentGTHROFileForEachBpel/ src/orabpel/ stagecurrentgthrofileforeach bpel/BIN0.java]:51 code too large`	Optimize the integration. For example, split the integration into multiple integrations.
If you attempt to activate an integration that includes a custom adapter, and the log file shows an `IllegalStateException` error, the metadata must be regenerated.	The activation issue is caused by mismatches in the metadata caused by the WSDL not being regenerated for some operations. The resolution is to update all operations directly or indirectly dependent on a resource when the metadata for that resource changes. For custom adapters, either click through the adapter in the Adapter Endpoint Configuration wizard to regenerate the metadata or perform the following action: 1. Open the `META-INF/cloud-adapter.xml` file. 2. Add the following information parallel to `sdkVersion`. If the `features` tag already exists, add this one line inside it. `<features> <feature enabled="true">sdk.wsdl.regeneration</feature> </features>`

Delay in Message Processing of First Instance after Activation

After activation of an integration, there is a several second delay between when the instance is created and when the message is processed. This is the expected behavior.

On the first request for all integrations, the artifacts and backend processes are lazy loaded as a first step. This can make for a longer time period spent between the first two steps shown in the activity stream when compared to subsequent invocations. This is the expected behavior for the initial request, and by design.

For example, you may see the following in the activity stream:

```
Sep 3rd 12:43:16.442 PM UTC 2020 Message received: Instance create
Sep 3rd 12:43:23 Message processed by Trigger
```

Note the seven second delay between when the message was received and processed.

A

Oracle Integration Use Cases

Use Oracle Integration in scenarios such as the following.

Topics:

Create an Integration to Exchange Messages Between Oracle Integration and a Local Host

In this tutorial, you'll learn how to transfer messages from Oracle Integration to a file on your local host using the connectivity agent.

In the first part of the tutorial, you'll download and install the connectivity agent in your local environment. Subsequently, you'll create a file-based connection to the agent, create an integration to write a message—received through a REST request—to a file on your local machine, and test the integration's output.

Upon completing this tutorial, you'll be familiar with several important skills related to the connectivity agent, such as creating an agent group, installing the connectivity agent, creating a connection using the agent, and configuring a simple integration to exchange messages between Oracle Integration and your system.

Topics:

Install the Oracle Integration Connectivity Agent

To exchange messages between Oracle Integration and a host machine, you must download and install the connectivity agent on the machine.

The following topics guide you through the process of installing the agent in your environment.

Topics:

- Download the Connectivity Agent Installer
- Install the Connectivity Agent on Your Local Host

Prerequisites to Download and Install the Connectivity Agent

To download and install the connectivity agent, you need:

- Access to Oracle Integration. Don't have access? Use free credits to try Oracle Integration now.
- Sign-in credentials (user name, password, data center/region, and identity domain) for your Oracle Integration user account.
- The Java Platform, Standard Edition 17 Development Kit (JDK 17) installed on your local host. See JDK Development Kit 17 Downloads. The use of JDK for the connectivity agent only is licensed as part of Oracle Integration usage.
- One of the supported operating systems on the host. See System Requirements.

> **Note:**
>
> This tutorial demonstrates installation of the connectivity agent on a Linux machine.

Create an Agent Group

The connectivity agent installation setup requires you to associate each agent installation with an agent group in Oracle Integration.

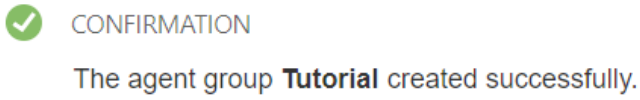
Let's create a new agent group for our installation. You can reuse this group for another installation of the agent on the same or a different host. An agent group supports up to two connectivity-agent installations, and an Oracle Integration instance supports up to five agent groups.

To create an agent group:

1. In the navigation pane, click **Integrations**, then click **Agents**.
2. Click **Create Agent Group**. The Create New Agent Group window is displayed.
3. Enter the following information and click **Create**.

Field	Information to Enter
Name	Enter `Tutorial`.
Identifier	Accept the default identifier value.
Description	Enter the following text: `Facilitates exchange of messages between Oracle Integration and a local host.`

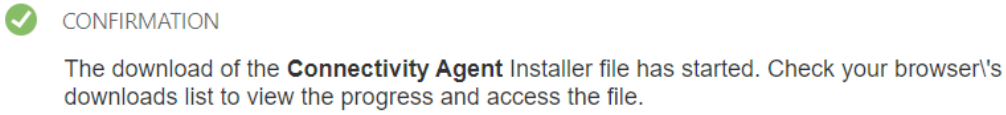
A confirmation message is displayed when a group is successfully created.



Download the Connectivity Agent Installer

On the Agents page, click **Download** and then **Connectivity Agent** to download the agent installer file to your machine.

A confirmation message informs that the download has begun.



Install the Connectivity Agent on Your Local Host

To run the agent installer file and perform other installation-related tasks, we'll use the command prompt.

> **Note:**
>
> The instructions and commands provided in this section apply to Linux machines and the Bash shell.

1. On your local machine, create a directory to install the connectivity agent. For example, run the following command from your home directory (that is, `/home/user1`).

   ```
   mkdir ConnectivityAgent
   ```

2. Move the installer file into the directory you just created. The following command shows the file being moved from the `Downloads` folder.

   ```
   mv /home/user1/Downloads/oic_conn_agent_installer.zip /home/user1/
   ConnectivityAgent/oic_conn_agent_installer.zip
   ```

3. Navigate to the `ConnectivityAgent` folder.

   ```
   cd ConnectivityAgent
   ```

4. Unzip the installer file. You'll find the executable and configuration files within it.

   ```
   unzip oic_conn_agent_installer.zip
   ```

or

```
jar xf oic_conn_agent_installer.zip
```

5. To view the extracted files, list all files in the folder.

```
ls
```

6. Now, open the `InstallerProfile.cfg` file, and add the Oracle Integration URL and the agent group identifier within it. This associates the agent installation with the agent group you created earlier.

```
vi InstallerProfile.cfg
```

Copy the URL and the identifier values to the respective fields in the command prompt as follows. To get the agent group identifier, click **View** on your agent group in the Oracle Integration instance.

```
oic_URL=https://oic_host:ssl_port
agent_GROUP_IDENTIFIER=TUTORIAL
```

Use the following key sequence to save and exit the configuration file: **Esc**, and then **Shift + z z**.

7. Provide execute permissions on all items within the `ConnectivityAgent` folder (including the executable file, `connectivityagent.jar`) for yourself (the owner) or for anyone else that may run the installation file. For example:
To provide all permissions (read, write, and execute) only to the owner, use:

```
chmod -R 700 .
```

To provide all permissions to everyone, use:

```
chmod -R 777 .
```

8. Append the `PATH` environment variable with the location of Java binaries. This helps you run Java executables (in this case, `connectivityagent.jar`) from any directory without having to type the full path of the binaries for each execution.
9. Check if the `PATH` variable has been updated successfully:

```
echo $PATH
```

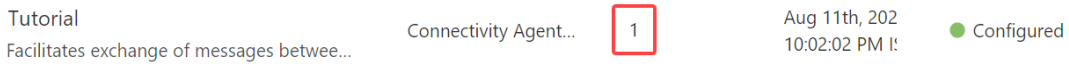
10. Run the `connectivityagent.jar` file to install the agent on your machine.
Enter the following command:

```
java -jar connectivityagent.jar
```

11. Enter your Oracle Integration credentials when prompted. After the installation completes, you'll see the following message:

```
Agent started successfully...Now available for new messages...
```

12. Refresh the Agents page on your Oracle Integration instance, and you'll find the agent you installed registered against the `Tutorial` group.



    Click the **Actions** menu of the agent group, and then click **Agents** to view the machine details on which the agent is running.

13. To stop the agent on your host, enter `crtl+c` in the command prompt. To restart it, use the same command provided in **Step 9**.

Congratulations! You've now successfully installed the Oracle Integration connectivity agent on your local host.

Use the Connectivity Agent to Configure an Example Integration

In this part of the tutorial, you'll create an integration that uses the connectivity agent and writes messages to files on your host machine.

The example scenario we'll use to create the integration has the following flow:

1. An external application sends a message through a POST request to Oracle Integration.
2. An app-driven integration with a REST interface trigger receives the message.
3. A File Adapter action in the integration writes this message to a file on a local machine using the connectivity agent.

Topics:

Prerequisites to Set Up the Integration

To complete this tutorial, you need:

- Access to Oracle Integration. Don't have access? Use free credits to try Oracle Integration now.
- Sign-in credentials (user name, password, data center/region, and identity domain) for your Oracle Integration user account.
- The Oracle Integration Connectivity Agent installed and running on your host machine. See Install the Oracle Integration Connectivity Agent.
- A directory on your host machine for the output files. See Create a Directory on the Host Machine for Output Files.

Create a Directory on the Host Machine for Output Files

On your local machine, create a separate directory to hold the output files from Oracle Integration. You'll provide the path of this directory while configuring the integration actions.

Navigate to your home directory, and create a directory named `output`.

```
mkdir output
```

Create a File Connection Using the Agent

You'll use a File Adapter in the integration to connect to your host machine and write messages to files. In order to connect to your machine, the File Adapter requires a connection set up using the connectivity agent you've installed.

To create a file connection:

1. In the navigation pane, click **Integrations**, then click **Connections**.
2. Click **Create**.
   The Create Connection — Select Adapter dialog is displayed.
3. Select the **File** from the dialog.
   The Create Connection dialog is displayed.
4. Enter the information to describe the connection.
   a. Enter a name for your connection (`File Connection`).
   b. Select **Trigger and Invoke** in the **Role** field.
   c. Enter an optional description for the connection.
5. Click **Create**.
   Your connection is created and you are now ready to configure other details.
6. In the Agent Group section, click **Configure Agents**, select **Tutorial**, and click **Use**.
7. Click **Save**.
8. Click **Test** to ensure that your connection is successfully configured.
   A confirmation message is displayed if your test is successful.

9. Click **Back** to return to the Connections page. Click **Save** again if prompted.
   Now you're ready to create the integration.

Create an App Driven Integration

Let's create an app-driven integration to receive a POST message from an external application and write the message to a file on your host machine.

1. In the navigation pane, click **Integrations**.
2. On the Integrations page, click **Create**.
   The Create Integration - Select a Style dialog is displayed.

3. Select **App Driven Orchestration**. The Create New Integration dialog is displayed.
4. Enter the following information.

Field	Information to Enter
What do you want to call your integration?	Provide a name for the integration, `Write Messages to Files`.
Identifier	Accept the default identifier value.
Version	Accept the default version number of `01.00.0000`.
What does this integration do?	Enter the following text: `Receives messages from a remote application and writes them to files on a host machine.`

5. Click **Create**. The integration canvas is displayed, where you can configure your integration.

Add a REST Interface to Receive Messages

Add a REST interface as a trigger connection to receive messages from an external application.

1. Click the **+** sign below **START** on the integration canvas to view a list of available trigger connections.
2. Select **Sample REST Endpoint Interface** from the list.
The Configure REST Endpoint wizard is displayed.
3. On the Basic Info page, enter a name without a space (`ReceiveMessages`) and a description for the endpoint. Click **Next**.
4. On the Resource Configuration page, enter the following details.

Field	Information to Enter
What is the endpoint's relative resource URI?	Enter `/msg`.
What action do you want to perform on the endpoint?	Select **POST**.
Add and review parameters for this endpoint	Select the check box.

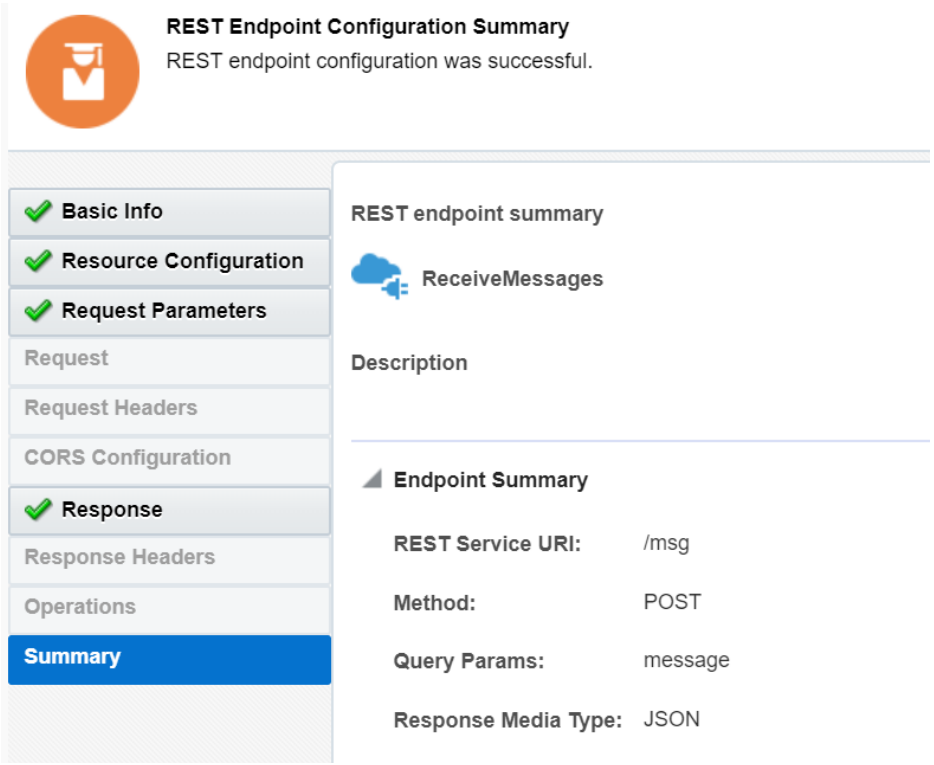
Field	Information to Enter
Configure this endpoint to receive the response	Select the check box.

Click **Next**.

5. On the Request Parameters page, specify a query parameter. This parameter will contain the message received from the external application. Click **+** in the Specify Query Parameters section to add a new row. Enter `message` in the **Name** field and select **string** in the **Data Type** field. Click **Next**.
6. On the Response page, select **JSON Sample** in the **Select the response payload format** field, and click the **inline** link to enter a sample JSON. Enter the following sample and click **OK**.

```
{
"msg": "Hello"
}
```

7. Click **Next**. On the Summary page, review the data you've entered and click **Done**.



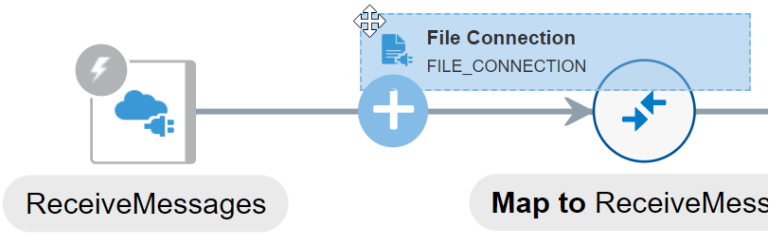
Note that a map action corresponding to the **ReceiveMessages** action is automatically added to the canvas. You'll configure it later in this tutorial.

8. In the canvas toolbar, select **Horizontal** from the **Layout** list. Click **Save**.

Configure a File Adapter to Write Messages to Files

Add a File Adapter to write messages received by the REST interface to files on your host machine.

1. Click **Invokes** on the pane to the right of the canvas.
2. Expand **File** and drag **File Connection** to the **+** sign after the **ReceiveMessages** action.



The Configure File Operation Endpoint wizard is displayed.

3. On the Basic Info page, enter a name without a space (`WriteFile`) and a description for the adapter. Click **Next**.
4. On the Operations page, enter the following details and click **Next**.

Field	Information to Enter
Select Operation	Select **Write File**.
Specify an Output Directory	Enter the path to the directory you created earlier. See Create a Directory on the Host Machine for Output Files. For example, your path can be `/home/user1/output`.
File Name Pattern	Specify the file name pattern for the new files that will be created to write messages. For example, enter the following pattern: `msg%SEQ%.json` This pattern creates a JSON file with the name `msg1.json` for the first message received, `msg2.json` for the second message, and so on.
Append to Existing File	Leave it unchecked.

5. On the Schema page, leave **Yes** selected in the first field, and in the second field, select **Sample JSON document**. Click **Next**.
6. On the File Contents - Definition page, click **Choose File** to upload a sample JSON document. Create a `.json` file with the name `sample.json`, add the following text to it, and upload the file on this page.

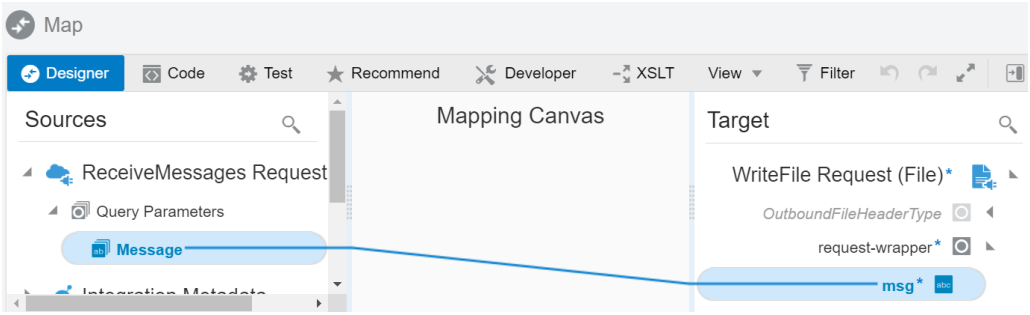
```
{
"msg":"Hello"
}
```

Click **Next**.

7. On the Summary page, review the data you've entered and click **Done**.
8. Note that the **Map to WriteFile** action is automatically added to the canvas. Click it and select **Edit** to define the mapping.

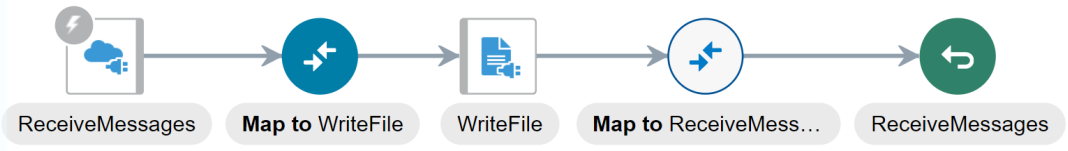
a. In the mapper, expand **Query Parameters** under **ReceiveMessages Request (REST)** on the left, and expand **request-wrapper** on the right.

b. Map **Message** on the left to **msg** on the right. This mapping ensures that the message received by the REST interface is passed on to the **WriteFile** action.



c. Click **Validate** and then **Close**.

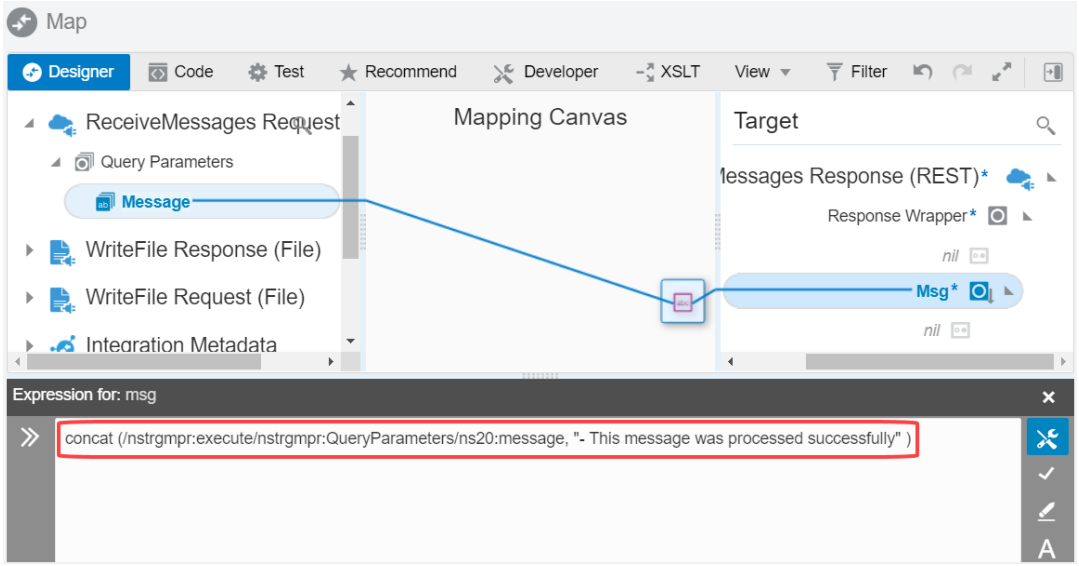
9. At this point, the integration flow appears as follows:



Configure Data Mapping to Return a Response

Configure the data mapping in the **Map to RecieveMessages** action to return a response to the external application after each message has been received and written to a file on the host machine.

1. Click the **Map to RecieveMessages** action and select **Edit** to define the mapping.
2. In the mapper, expand **Query Parameters** under **ReceiveMessages Request (REST)** on the left, and expand **Response Wrapper** on the right.
3. Map **Message** on the left to **Msg** on the right.
   The expression builder with a corresponding mapping expression opens.
4. Click **Toggle Functions** in the toolbar to view the Components pane. Expand **String** and drag the **concat** function to the expression builder pane.
5. Copy the existing expression in the pane (for example, `/nstrgmpr:execute/nstrgmpr:QueryParameters/ns20:message`) into the concat function as the first operator and add `"- This message was processed successfully"` as the second operator. The final expression in the pane will be of the following format: `concat ( /nstrgmpr:execute/nstrgmpr:QueryParameters/ns20:message , " - This message was processed successfully" ).`
   This mapping ensures that a success message is sent back to the external application after each incoming message is written to a file.
6. Click **Save** and click **Validate**.
7. Click **Close** to return to the integration canvas.



Activate and Test Your Integration

Check for errors, save, and activate the integration flow.

You'll notice an error notification on the canvas. To resolve it, assign a primary business identifier for tracking. Business identifiers enable you to track payload fields in messages during runtime. A primary business identifier is required to activate an integration. See Assign Business Identifiers for Tracking Fields in Messages.

To assign an identifier:

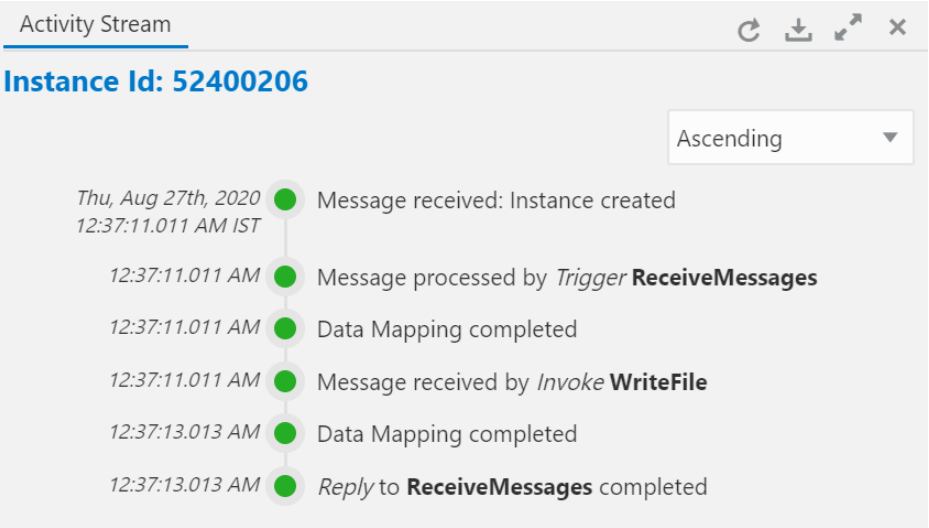
1. Click **Actions Menu** in the top-right corner of the canvas, and select **Tracking**.
2. On the resulting page, select **message** on the left and move it to the table on the right.
3. Click **Save**.
4. Save the integration and click **Close**.

Now, activate and test your integration.

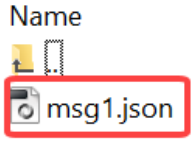
1. On the Integrations page, click the **Activate** button against your integration.
2. In the Activate Integration dialog, select **Enable Tracing** and **Include Payload** check boxes, and click **Activate**.
3. After the integration is activated, click **Run** and select **Test** to test run the integration.
4. On the resulting page, enter a message (for example, `Hello World`) to send to the integration as a POST request, and click **Test**.
5. After the message has been processed by the integration and written to a file on your host machine, you receive the following response on the Test page:

```
{
  "msg" : "Hello World - This message was processed successfully"
}
```

The activity stream also appears on the page, showing the successful execution of the integration and all the actions within it.



6. On your host machine, navigate to the `output` folder to find the file written by the integration. Open the file and verify its contents.



7. Repeat **Step 4** to write another file on the host machine.

Congratulations! You've successfully created an integration using the connectivity agent and transferred messages from Oracle Integration to a local host.

Read Large XML Files Containing Multiple Namespaces

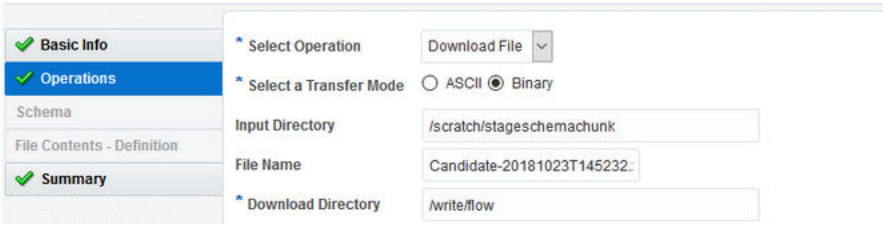
You can use a stage file action to read a large XML file that contains multiple namespaces. You upload a ZIP archive file containing the contents. While this use case describes how to use the FTP Adapter with the stage file action, other adapters are capable of receiving an XML document as an attachment in the response (for example, the SOAP Adapter, REST Adapter, and Oracle Taleo Enterprise Edition Adapter).

The following XML example shows an export of a candidates list obtained from Oracle Taleo Enterprise Edition. The XML file has a repeating structure and contains more than one namespace in the document.

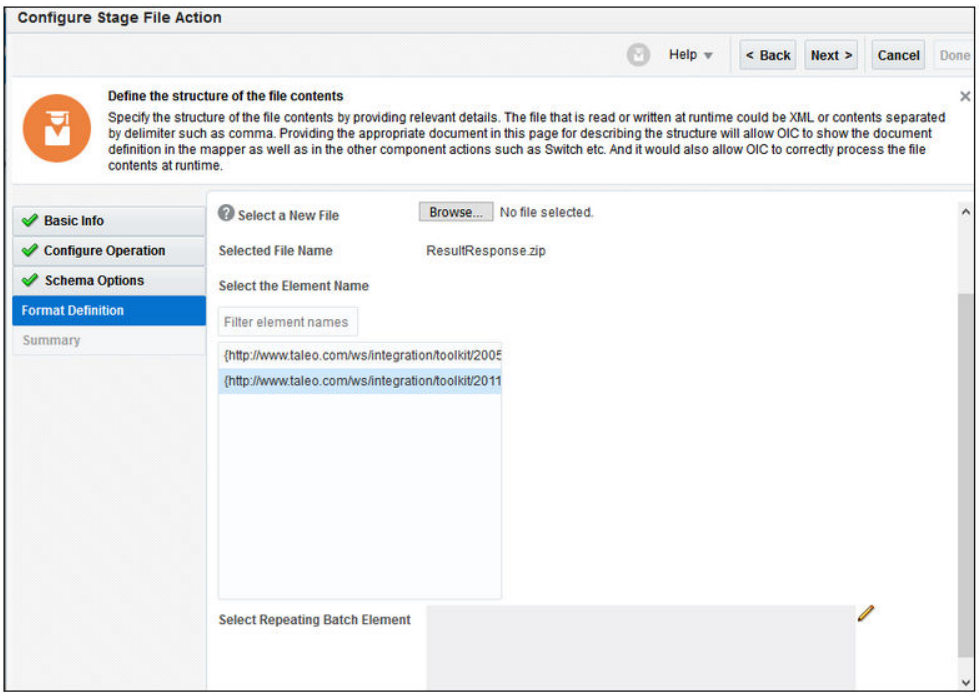
```
<Document xmlns="http://www.taleo.com/ws/integration/toolkit/2011/05">
    <Attributes>
        <Attribute name="count">3327</Attribute>
        <Attribute name="duration">0:00:01.443</Attribute>
        <Attribute name="entity">Candidate</Attribute>
        <Attribute name="mode">XML</Attribute>
        <Attribute name="version">http://www.taleo.com/ws/tee800/2009/01</Attribute>
    </Attributes>
    <Content>
        <ExportXML xmlns="http://www.taleo.com/ws/integration/toolkit/2005/07/action/export">
            <record>
                <field name="Number">10160</field>
                <field name="FirstName">Performance</field>
                <field name="LastName">HR Admin</field>
                <field name="EmailAddress">hradmin@invalidmail.com</field>
                <field name="CreationDate">2011-11-16T14:36:14-05:00</field>
            </record>
            <record>
                <field name="Number">10163</field>
                <field name="FirstName">Performance</field>
                <field name="LastName">Manager 1</field>
                <field name="EmailAddress">manager1@invalidmail.com</field>
                <field name="CreationDate">2011-11-16T14:42:10-05:00</field>
            </record>
            <record>
                <field name="Number">10261</field>
                <field name="FirstName">Performance</field>
                <field name="LastName">Matrix Manager</field>
                <field name="EmailAddress">tpmatrix@invalidmail.com</field>
                <field name="CreationDate">2012-03-12T08:28:44-04:00</field>
            </record>
            <record>
```

This use case provides a high level overview of the steps to read a large XML file with multiple namespaces in segments in an integration. Two FTP Adapters are used for downloading a large ZIP file from a remote FTP server and writing the staging file to a remote FTP server.

1. Add an FTP Adapter to a scheduled orchestrated integration.
2. Configure the FTP Adapter with the **Download File** operation and specify the input directory, file name, and ZIP file download directory.



3. Add a stage file action to the integration and configure it as follows:
   a. On the Configure Operation page, configure the following options:
      - Select **Read File in Segments** from the **Choose Stage File Operation** list.
      - Select **Yes** for **Configure File Reference**.
      - Specify the file reference for the ZIP file downloaded from the FTP server in the **Specify the File Reference** section.
   b. On the Schema Options page, select **XML schema (XSD) document**.
   c. On the Format Definition page, configure the following options:
      - Click **Browse** to upload the archive of the schemas (XSDs) that represents the message structure.
      - Select the root element that contains the repeated element.

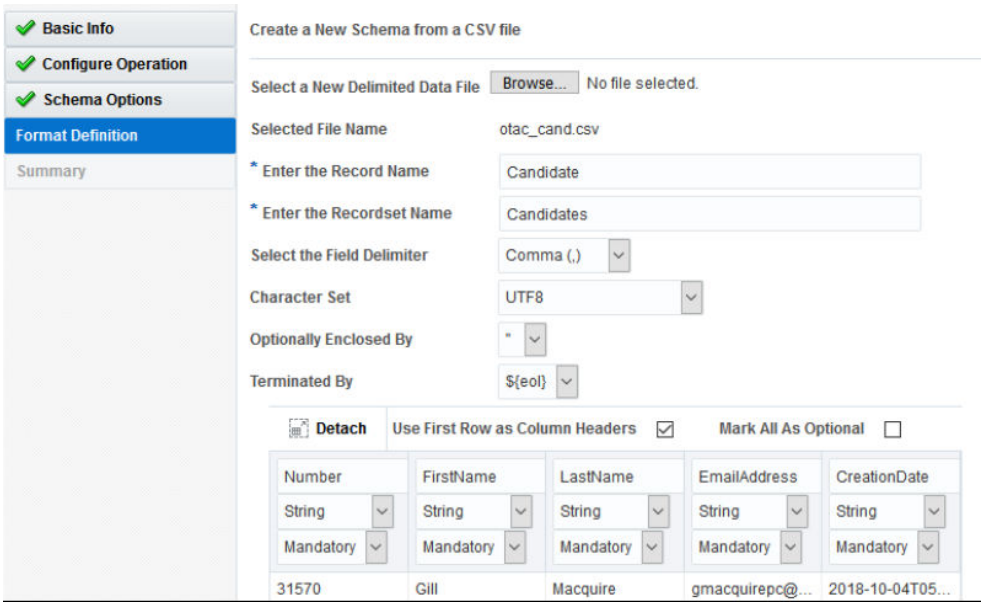


- Click the **Edit** icon to select the repeating batch element in the Expression Builder.

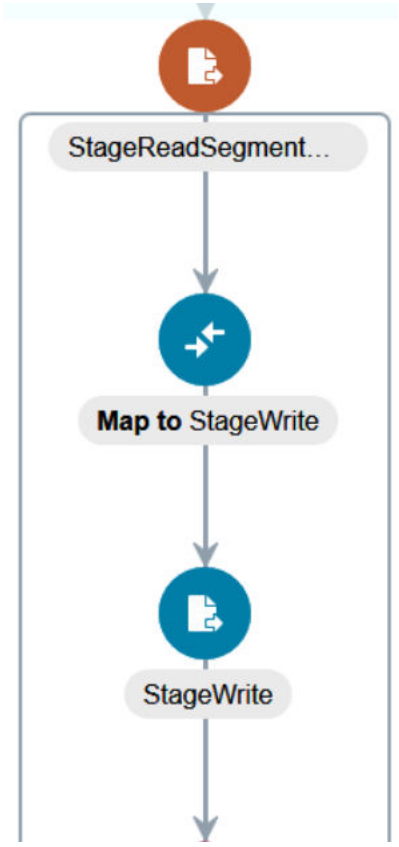
The Summary page looks as follows.

4. Add a second stage file action to write the records in comma-separated value (CSV) format.
   a. On the Configure Operation page, configure the following options:
      - Select **Write File** from the **Choose Stage File Operation** list.
      - Specify the file name of the CSV file in the **Specify the File Name** section.
      - Specify the output directory in the **Specify the Output Directory** section.
      - Select the **Append to Existing File** checkbox.
   b. On the Schema Options page, select **Sample delimited document (e.g. CSV)**.
   c. On the Format Definition page, configure the following options:

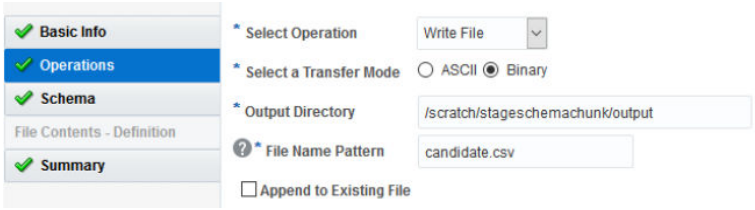
- Click **Browse** to select the sample CSV file.
- Specify the record and record set name.



5. Perform the mapping between the two stage file actions.



6. Add a second FTP Adapter to write the staging file to an output directory on a remote FTP site.



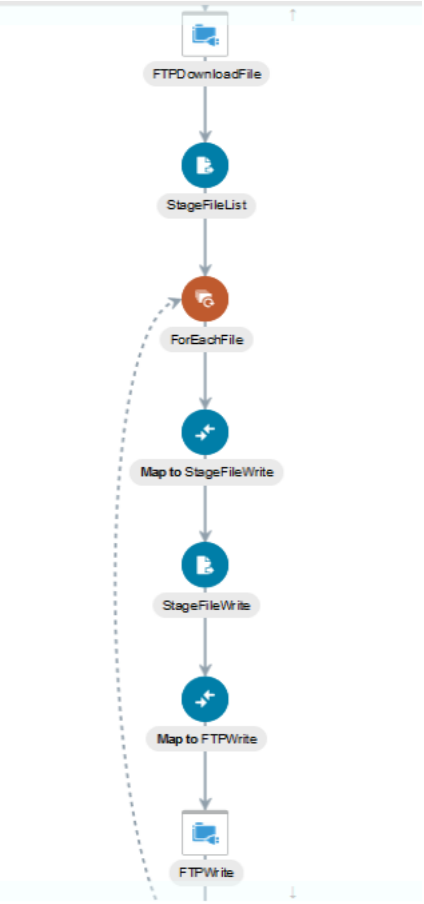
7. Design the remaining parts of the integration.

Use a Stage File Write Operation with an Opaque Schema

You can use an opaque schema in a stage file action Read File or Write File operation without concern for a schema for the file. The only condition is that whatever is sent to the opaque element in the opaque schema must be base64-encoded data.

For the following use case:

- The integration downloads a zip file using the FTP Adapter Download File operation and unzips and processes each file.
- The files are staged temporarily in Oracle Integration using a stage file action List Files operation.
- The files are iterated over using a for-each action.
- The files are written to a staged directory using a stage file action Write File operation with an opaque schema.
- The files are written to an output directory using the FTP Adapter Write File operation.

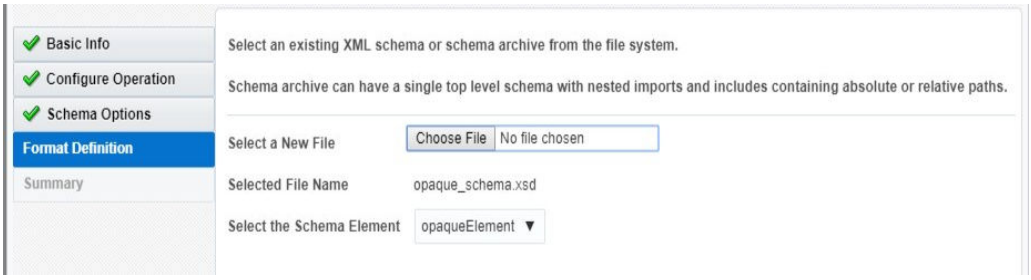


1. Add a stage file action to an orchestrated integration. See Configure a Stage File Action.
2. On the Configure Operation page, select the **Write File** operation and specify the file name and output directory.
3. On the Schema Options page, select **XML schema (XSD) document**.
4. On the Format Definition page, make the following selections.
   a. Select a new file (for this example, **opaque_schema.xsd** is selected).

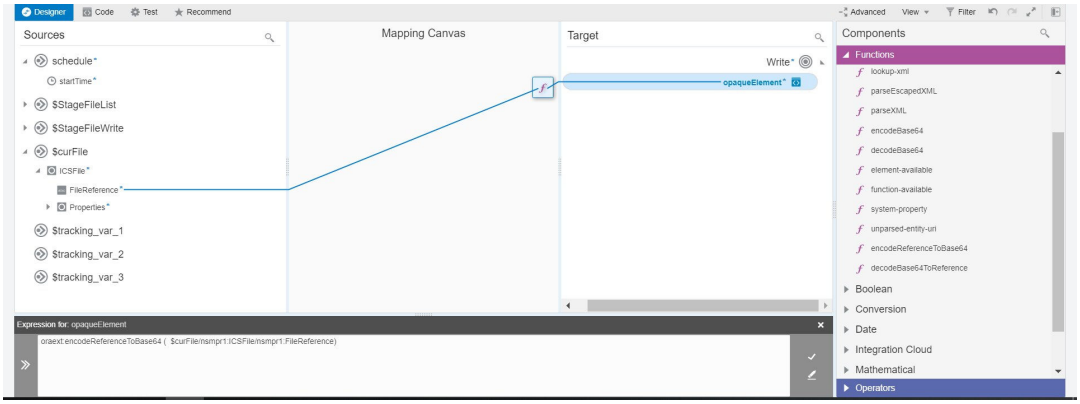
      The contents of the XSD file are as follows for this example:

      ```
      <?xml version = '1.0' encoding = 'UTF-8'?>
      <schema targetNamespace="http://xmlns.oracle.com/pcbpel/adapter/opaque/"
                  xmlns="http://www.w3.org/2001/XMLSchema" >
            <element name="opaqueElement" type="base64Binary" />
          </schema>
      ```

   b. Select the schema element (for this example, **opaqueElement** is selected).



5. Map the **FileReference** of the for-each repeating element (**ICSFile**) to **opaqueElement** of **StageFileWrite** using an **encodeReferenceToBase64** function.



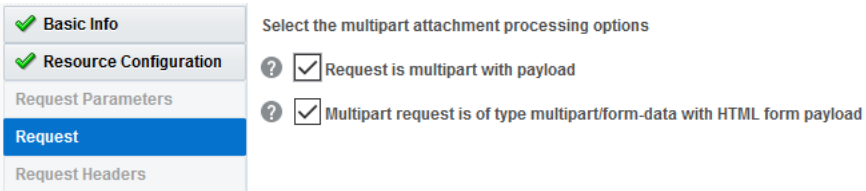
Test REST Adapter Trigger-Based Integrations with Multipart Attachments on the Test Page

You can test REST Adapter trigger-based integrations with multipart attachments on the Test page. This section provides high-level use cases for testing the following multipart attachment types.

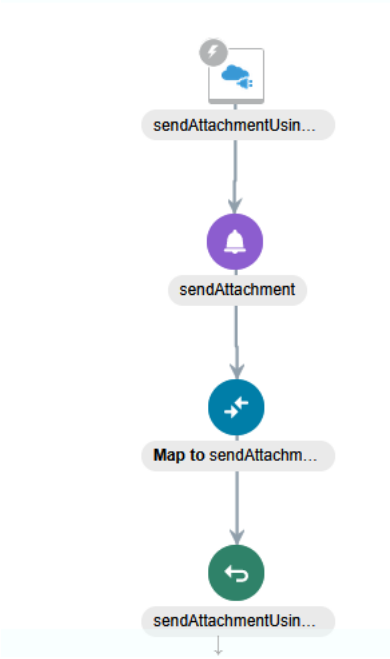
- multipart/form-data
- multipart payload with structured data (for this example, JSON)

Test Multipart/Form-Data

1. Configure a REST Adapter as a trigger connection and select the following options on the Request page:
   - **Request is multipart with payload**
   - **Multipart request is of type multipart/form-data with HTML form payload**

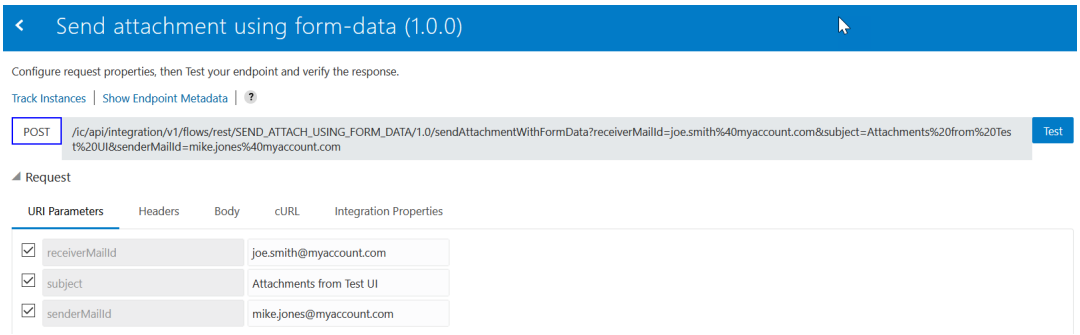


2. Design the remaining parts of your integration. For this example, the integration includes a REST Adapter trigger connection, notification action, mapper, and return action.

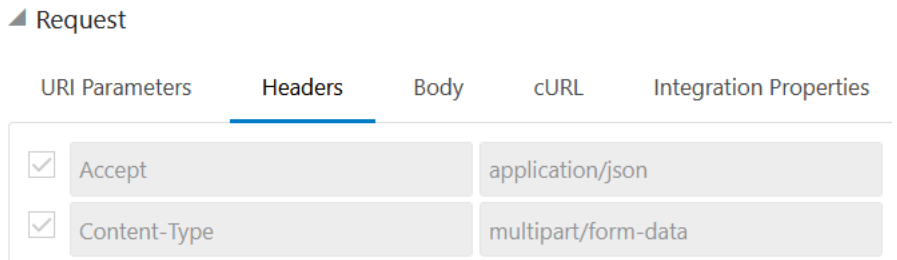


3. Activate the integration.
4. Click

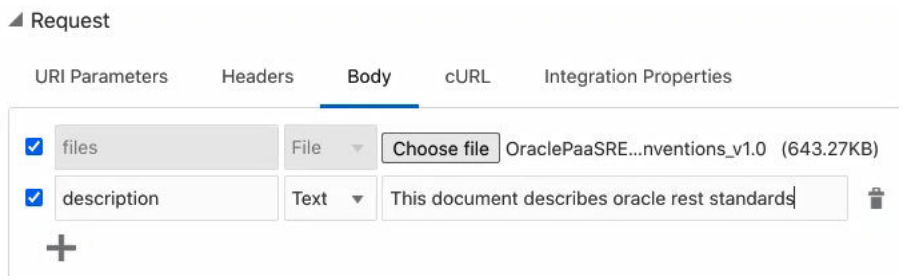
, then click **Test** on the message that is displayed.
5. In the **URI Parameters** section of the Test page, specify values for the parameters.



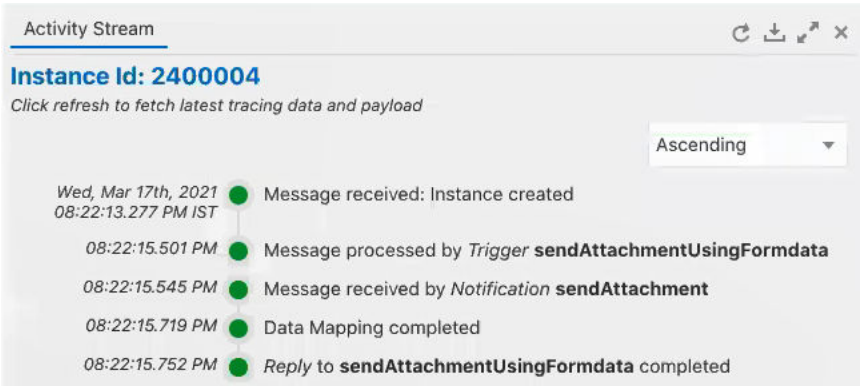
6. In the **Headers** section, view the header details.



7. In the **Body** section, select the attachments file to upload and specify a text description for the email.



8. Click **Test** to invoke this REST Adapter-triggered integration.
The activity stream results are displayed on the right side of the page.

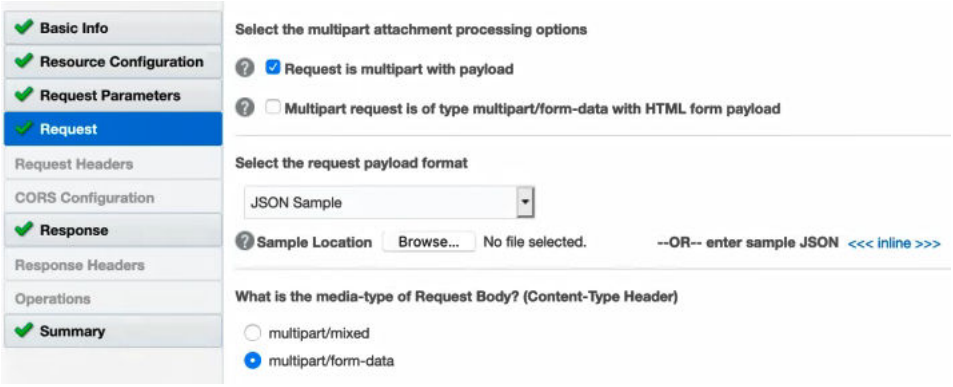


9. Check your response email for the file attachment you selected and the text description you specified.

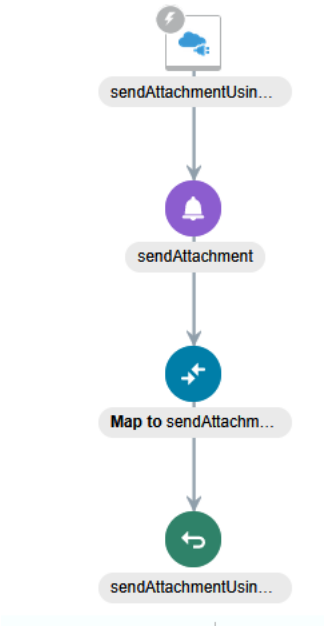
```
This document describes oracle rest standards
```

Test Multipart Payload with JSON Structured Data

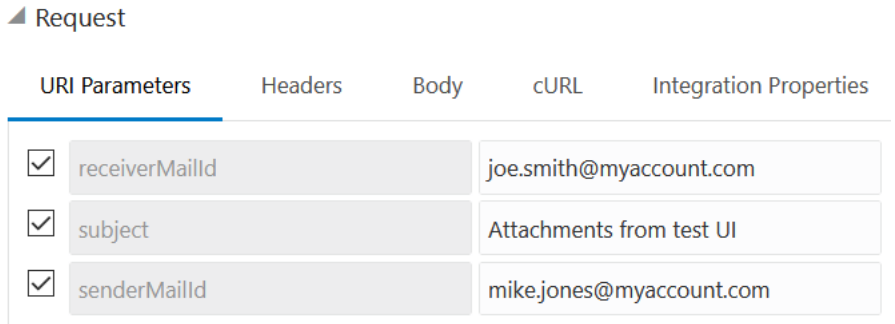
1. Configure a REST Adapter as a trigger connection and select the following options on the Request page:
   - **Request is multipart with payload** as the multipart attachment processing option.
   - **JSON Sample** as the request payload format.
   - **multipart/mixed** or **multipart/form-data** as the media type of the request body. If you select **multipart/form-data**, then the maximum allowed attachment size for testing is 50 MB.



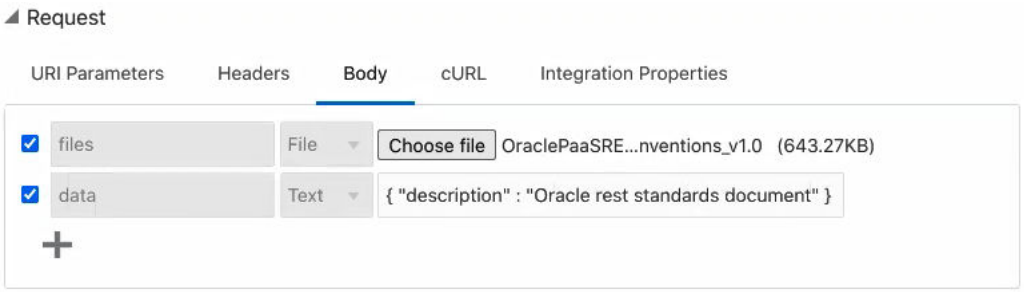
2. Design the remaining parts of your integration. For this example, the integration includes a REST Adapter trigger connection, notification action, mapper, and return action.



3. Activate the integration.
4. Click
▶
, then click **Test** on the message that is displayed.
5. In the **URI Parameters** section of the Test page, specify values for the parameters.



6. In the **Body** section, select the attachments file to upload and specify a text description for the email.



7. Click **Test** to invoke this REST Adapter-triggered integration.
The activity stream results are displayed on the right side of the page.

8. Check your response email for the file attachment you selected and the text description you specified.

```
Oracle rest standards document
```

The Test page provides a variety of testing features. See Test REST Adapter Trigger Connection-Based Integrations.

Calculate the Sum of a Column or the Count of Rows while Processing Large Files

You can process large comma-separated value (CSV) files (up to 1 GB in size) using the Read File In Segments operation of a stage file action. You may also have a requirement to calculate the sum of a column or the count of rows.

Consider the following payload. Assume you want to calculate the sum of all the values in the column **Amount1**. You may typically declare a variable upstream of the stage file action and keep updating this variable with the computation done in each chunk of the stage file action.

Sno	Amount1	Amount2	Amount3	Amount4
Integer	Integer	Integer	Integer	Integer
Mandatory	Mandatory	Mandatory	Mandatory	Mandatory
1	100	100	10.0	600
2	200	100	10.1	500
3	300	100	10.2	100
4	400	100	10.3	200

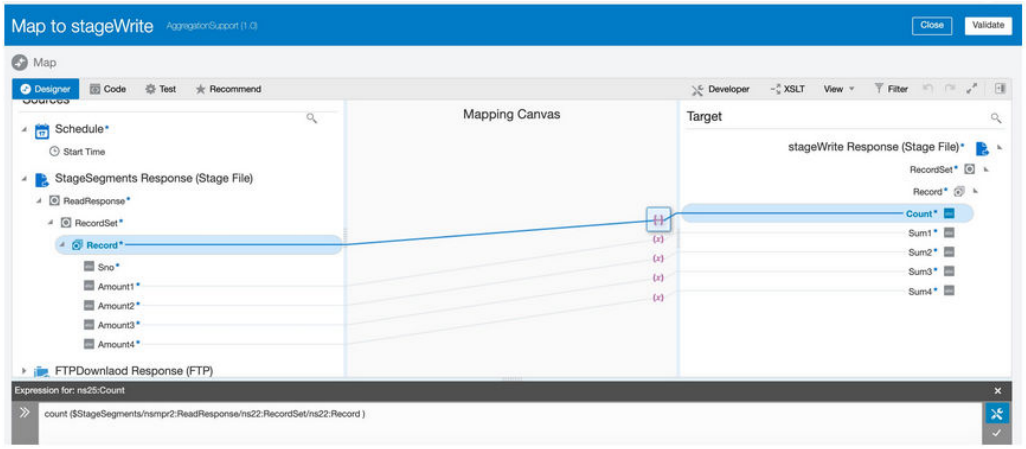
However, updating the upstream variables inside the stage file action Read File in Segments operation impacts performance and does not allow stage file action processing in parallel. You observe the following warning message in the integration canvas:

```
Stage File Read File in Segments includes action that will result in segments
being processed sequentially
```

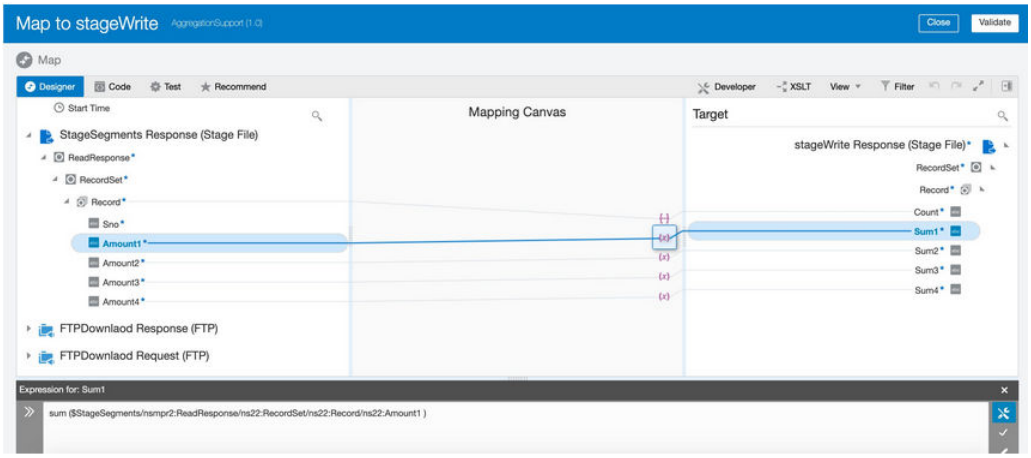
As a solution for this use case, perform the following steps to use the aggregate functions sum and count while processing larger files:

1. Add a stage file action in the integration canvas and select the Read File in Segments operation to read the large file.
2. Inside the stage file action that uses the Read File In Segments operation, create a second stage file action that uses the Write File operation and select the **Append to Existing File** check box.
3. Get the count and sum of each segment and write the results into the second stage file action.

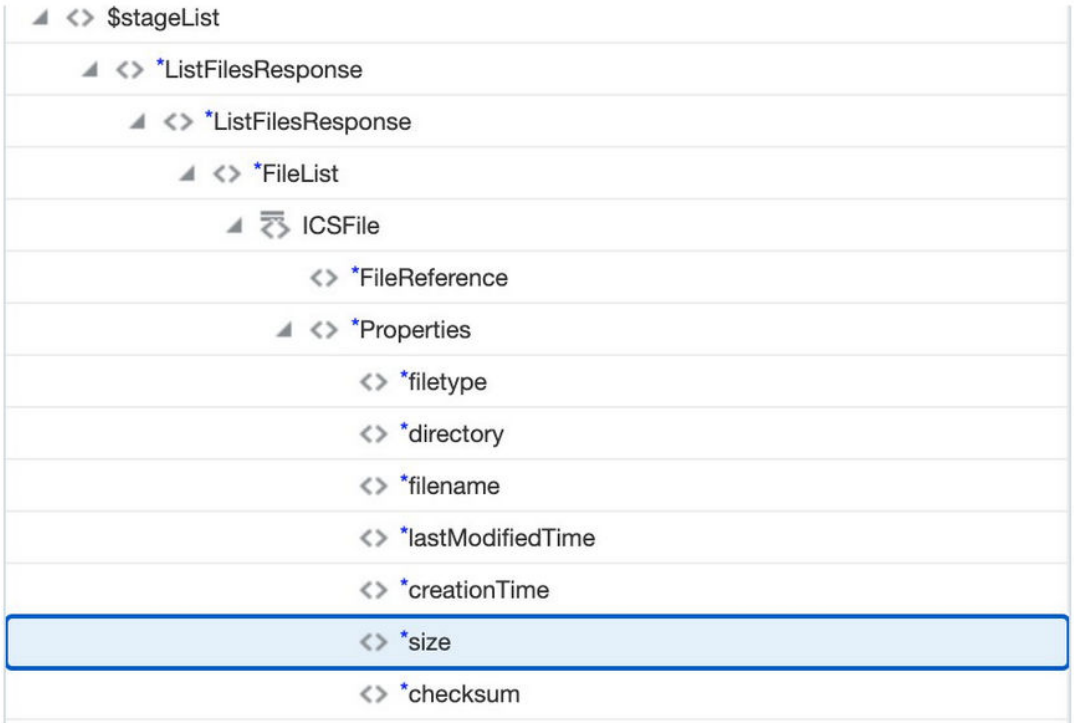
    - `count ($StageSegments/.../Record)`



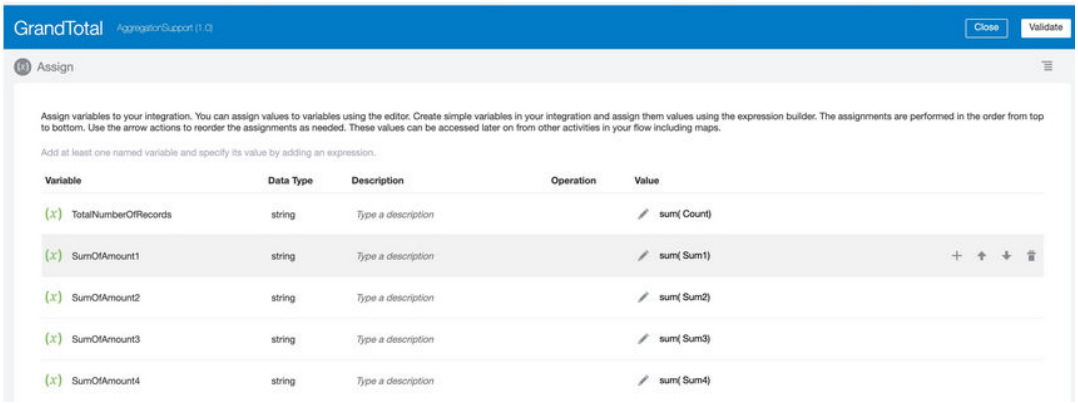
    - `sum ($StageSegments/..Record/ns22:Amount1)`



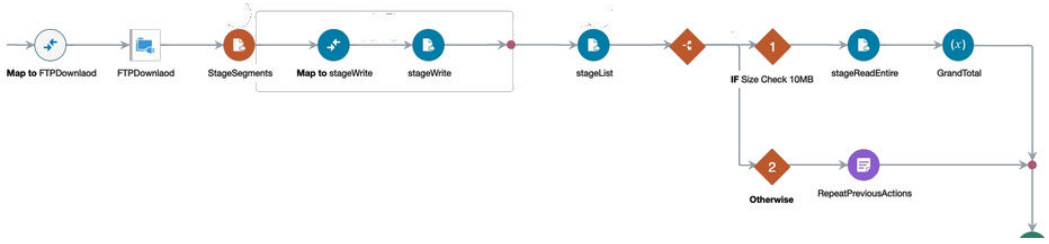
4. After completion of the stage file action using the Read File in Segments operation, create a stage file action that uses the List File operation to list the file that was created in Step 2. The response of the stage file action List File operation contains the file size. The file size determines the next action to perform.



5. If the file size is less than 10 MB, read the file with a stage file action that uses a Read Entire File operation and create an assign action with variables to store the grand totals.



6. Otherwise, repeat Step 1 through Step 4 with the new file.



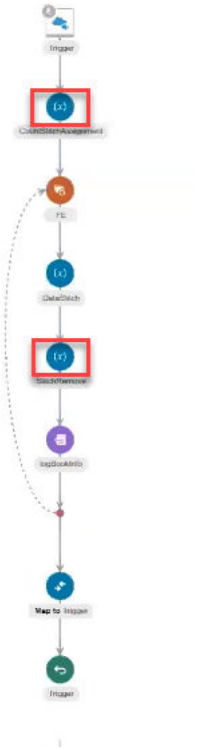
Stitch Editor Use Cases

Review the following stitch editor use cases.

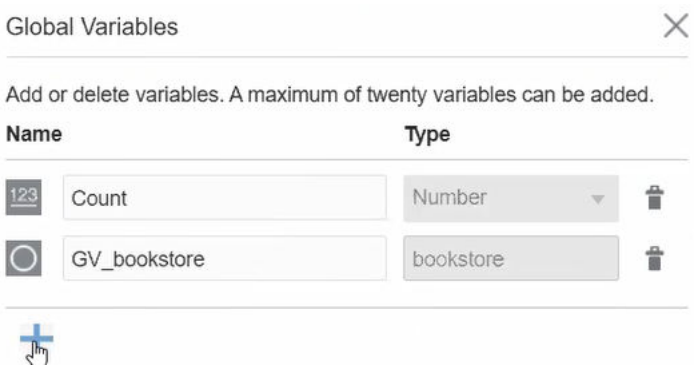
Topics:

Use the Three Stitch Operations (Assign, Remove, and Append) in an Integration

The following use case shows an integration that includes two stitch actions. All three stitch operations (assign, remove, and append) are used in this integration. This use case also describes how global variables are created and added to a stitch action.



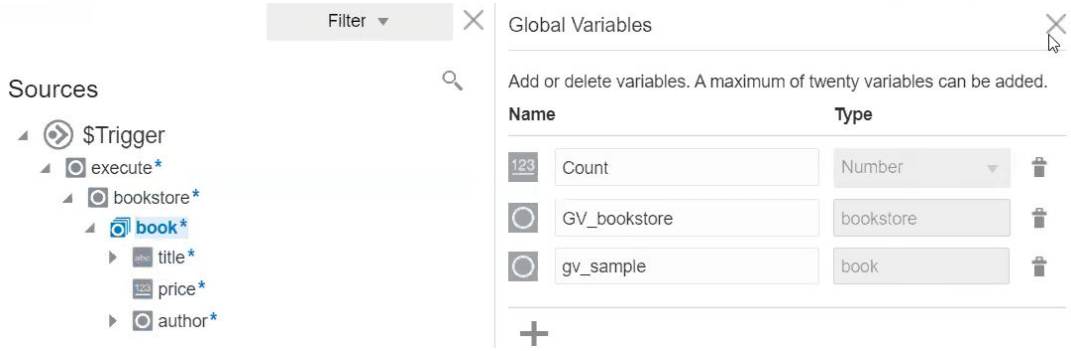
1. In the integration canvas, click
   (x)
   . For this example, two global variables are defined for the stitch action to use:
   - The **Count** scalar variable maintains the count in a for-each loop in the integration.
   - The **GV_bookstore** complex variable stores book information.



2. If you want to create additional global variables, click the **plus** sign.
3. From the **Type** list, select **Object** to define a complex type variable. All other available selections (**String**, **Boolean**, and so on) are used for creating scalar type variables.

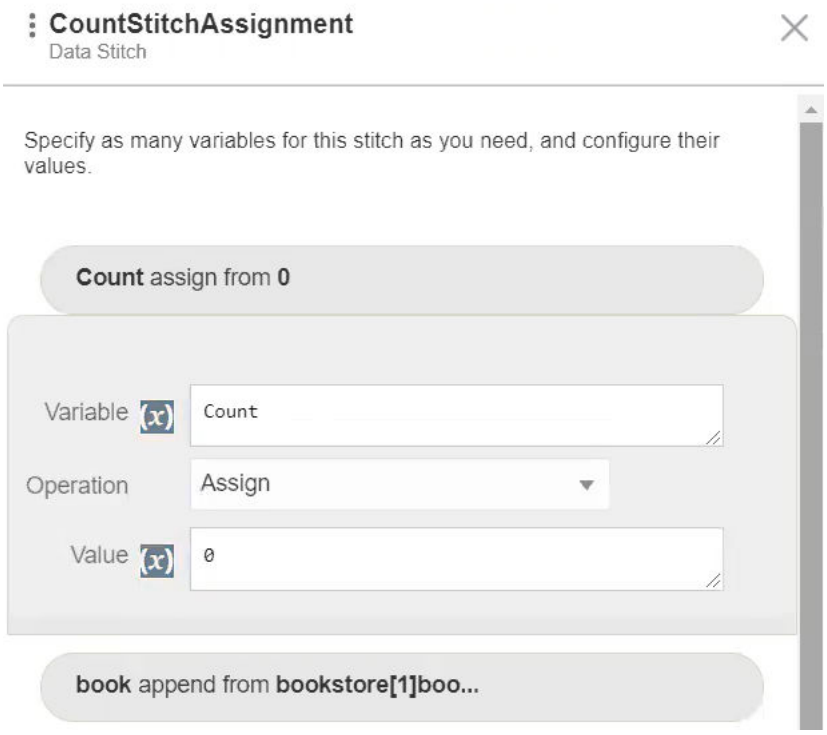
The **Sources** schema tree is displayed to show all data structures used in the integration.

4. Name the variable (for this example, **gv_sample** is used) and expand the schema tree to select the element to use (for this example, **book** is selected). You can also drag **book** to the **Type** field of **gv_sample**.

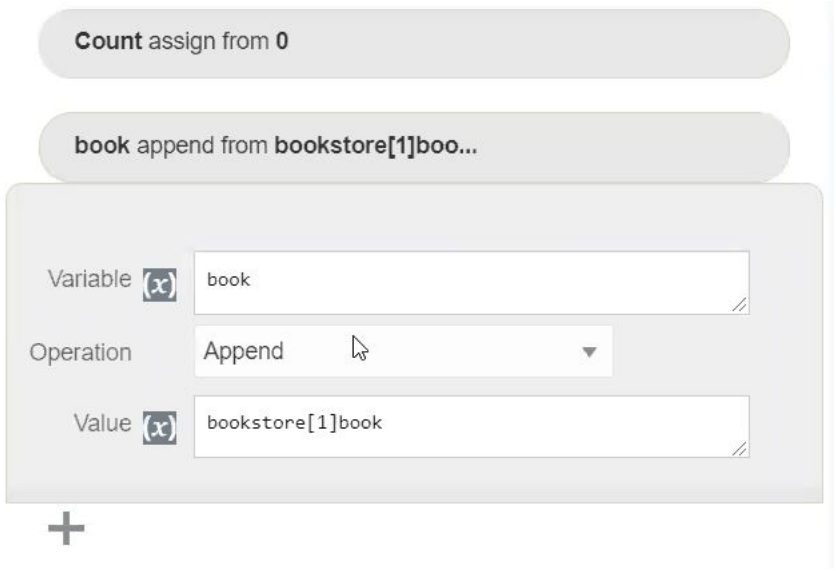


You are now ready to define complex assignments in the stitch editor.

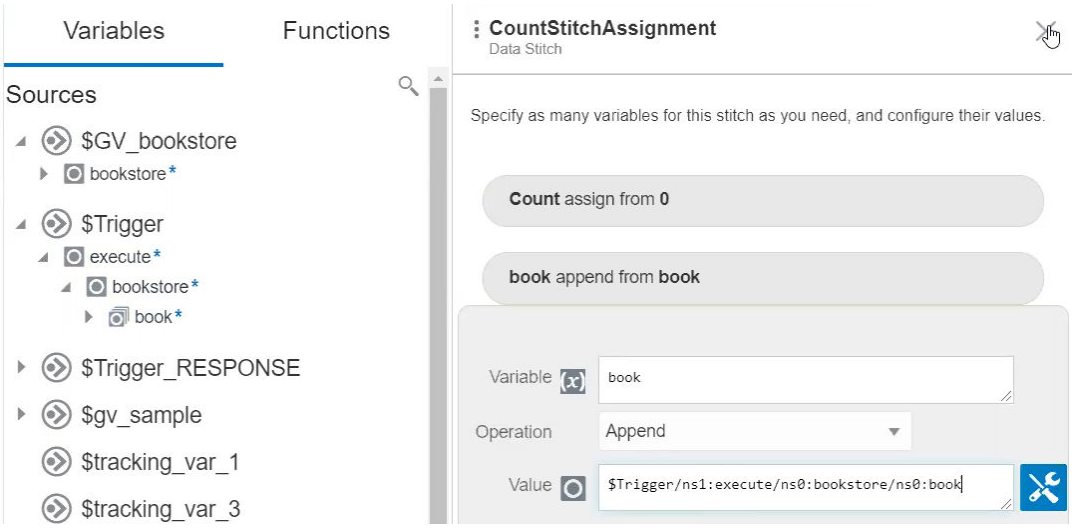
5. Return to the integration and open the first stitch action.
Note that two assignments are defined to perform different operations for this example.
   - The first assignment performs an assign operation. The **Count** variable to use in the for-each loop is initialized to zero.



- The second assignment performs an operation to append new book information to the existing book list.



6. Click to view the complete XPath expression of the **book** variable. The book information is appended to the book list.



7. Click **X** in the upper right corner to close the first stitch action.
8. Click the second stitch action, which is executed inside the for-each loop. This stitch action removes price information from each book during the looping iteration. The count index is executed in the for-each loop until the entire book list is read.

9. Activate and invoke the integration. For this example, the payload removes the price details and appends additional author details (for this example, `Joe Smith`).

```
<?xml version="1.0" encoding="utf-8"?>
<bookstore xsl:noNamespaceSchemaLocation="schema.xsd" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance">
 <book>
   <title lang="en">Java</title>
   <price>100USD</price>
   <author>
     <firstName>Joe</firstName>
     <lastName>Smith</lastName>
   </author>
 </book>
</bookstore>
```

10. View the results of the integration execution in the activity stream.

TO Element Does Not Exist in the Variable DOM

The following `TO` element use cases and their expected results are described.

ASSIGN into an Unbounded Element that Does Not Exist in the TO DOM

- Expected Result: The `TO` element is created and the `FROM` element is assigned.
- Assignment: `($OutputVarNameOnEB/ns0:root/ns0:b[1]/ns0:bch1).ASSIGN($V1/ns0:root/ns0:b[1]/ns0:bch1)`

$OutputVarNameOnEB before assign	$V1Content	$OutputVarNameOnEB Result

`<root xmlns="uri://ics/examples">`
`<a>I am original</a>`
`<d>`
`<dch1>I am original too</dch1>`
`</d>`
`</root>`
	`<root xmlns="uri://ics/examples">`
	`<b>`
	`<bch1>79</bch1>`
	`</b>`
	`</root>`
		`<root xmlns="uri://ics/examples">`
		`<a>I am original</a>`
		`<b>`
		`<bch1>79</bch1>`
		`</b>`
		`<d>`
		`<dch1>I am original too</dch1>`
		`</d>`
		`</root>`

APPEND a List into an Unbounded Element that Does Not Exist in the TO DOM

- Expected Result: The `FROM` list is inserted in the `TO` DOM at the right location, conforming with the schema.
- Assignment: `($OutputVarNameOnEB/ns0:root/ns0:b[1]/bch3).APPEND($V1/ns0:root/ns0:b[1]/ns0:bch3)`

$OutputVarNameOnEB before assign	$V1Content	$OutputVarNameOnEB Result
<root xmlns="uri://ics/examples"> <a>I am original <d> <dch1>I am original too</dch1> </d> </root>	<root xmlns="uri://ics/examples"> <bch3> <w>1.00</w> </bch3> <bch3> <y>I'm second</y> </bch3> <bch3> <w>3</w> <y>3rd</y> </bch3> <bch3> <w>99.99</w> </bch3> </root>	<root xmlns="uri://ics/examples"> <a>I am original <bch3> <w>1.00</w>< </bch3> <bch3> <y>I'm second</y> </bch3> <bch3> <w>3</w> <y>3rd</y> </bch3> <d> <dch1>I am original too</dch1> </d> </root>

Predicate Usage

The following predicate use cases and their expected results are described.

Assign or Append - The FROM Predicate Selects Zero Nodes (Out of Bounds)

- Expected Result: An exception is thrown at runtime. At design time, the system cannot detect if the predicate can select data.
- Runtime Error Assignment Example: `($t1).ASSIGN($v1/globalElement/element2[count(.)+1])`

> **Note:**
>
> The `[count(.)+1]` predicate is always out of bounds.
> This assignment fails at runtime and an exception is thrown.

Append to an Indexed TO - The TO Element Has a Predicate for an APPEND Operation - Not Supported

- Expected Result: The system at design time does not enable you to predicate the repeating element where the `APPEND` happens (last step on the `TO`). The `APPEND` always appends at the end of the list.

- Example: `($OutputVarNameOnEB/client:processResponse/client:c[2]).APPEND($InputVarNameOnEB/client:process/client:c)`
  - Not Supported: `"Predicated TO".APPEND()` (Assignment example 1)

    ```
    ($<OutputVarNameOnEB>/client:processResponse/client:c[@attr1='some existing value']).APPEND($InputVarNameOnEB/client:process/client:c[1])
    ```

    > **Note:**
    >
    > The `[@attr1='some existing value']` predicate filters/selects a subset of nodes. Appending into a subset of nodes makes no intuitive sense.

  - Not Supported: `"APPEND After"` (Assignment example 2)

    ```
    ($OutputVarNameOnEB/client:processResponse/client:c[2]).APPEND($<InputVarNameOnEB>/client:process/client:c)
    ```

    > **Note:**
    >
    > The `[2]` predicate is not intuitive. Do you `APPEND` to single instance 2, `APPEND` after instance 2, or `APPEND` before instance 2.

    Design-time error: The system throws an error at design time for both examples.

Assign to Predicated TO - The Predicate References an Existing Node

- Expected Result: The assignment is done into the `TO` element referenced by the predicate.

Assign to Indexed TO - The TO Indexed Predicate References a Node that Does Not Exist

- Expected Result: The system creates the missing nodes, but the missing nodes are created empty.
- Example: `($v1/ns0:po/ns0:header/ns0:customer/ns0:email[2]).ASSIGN("jim@yahoo.com")`

  > **Note:**
  >
  > The indexed predicate refers to a predicate whose execution resolved into a positive number.

- Missing `TO` Nodes Assignment Example: `($OutputVarNameOnEB/client:processResponse/client:c[count(.)+4]).assign($InputVarNameOnEB/client:process/client:c[1])`

> **Note:**
>
> The `[count(.)+4]` predicate is always out of bounds.

$OutputVarNameOnEB before assign	$InputVarNameOnEB Content	$OutputVarNameOnEB Result
<processResponse xmlns="http://xmlns.oracle.com/SOA/ComplexAssignment/BPELProcess1"> <a/> <x/> <c attr1="Old Value" attr2="Old Value"> <x>Old Value</x> <y>Old Value</y> <z>Old Value</z> </c> <d/> <e/> </processResponse>	<c attr1="New Value" attr2="New Value"> <x>New Value</x> <y>New Value</y> <z>New Value</z> </c>	<processResponse xmlns="http://xmlns.oracle.com/SOA/ComplexAssignment/BPELProcess1"> <a/> <x/> <c attr1="Old Value" attr2="Old Value"> <x>Old Value</x> <y>Old Value</y> <z>Old Value</z> </c> <c/> <c/> <c/> <c attr1="New Value" attr2="New Value"> <x>New Value</x> <y>New Value</y> <z>New Value</z> </c> <d/> <e/> </processResponse>

Assign to Indexed Ancestor TO - The TO Indexed Predicate References a Node that Does Not Exist

- Expected Result: The system creates the ancestor instances up to the index if they do not exist and then performs the assignment.
- Example: `($v1/ns0:po/ns0:details/ns0:item[$i]/itemName).ASSIGN("iPhone X") ...whereas $i is a number having value 4`

$v1 before assign	$v1 after assignment
<ns0:po> <ns0:details> <ns0:item code='ip100'> <ns0:itemName>iPad</ ns0:itemName> </ns0:item> <ns0:details> </ns0:po>	<ns0:po> <ns0:details> <ns0:item code='ip100'> <ns0:itemName>iPad</ ns0:itemName> </ns0:item> <ns0:item /> <ns0:item /> <ns0:item code='ip100'> <ns0:itemName>iPhone X</ ns0:itemName> </ns0:item> <ns0:details> </ns0:po>

Assign - The TO Logical Predicate References a Node that Does Not Exist

- Expected Result: An exception at runtime is thrown.
- Definition: A logical predicate is a predicate whose execution resolves into a boolean result: true or false. This case refers to when the result becomes false. Therefore, there is no `TO` element selected.
- Runtime Error Assignment Example: `($OutputVarNameOnEB/client:processResponse/client:c[@attr1='eng']).assign($InputVarNameOnEB/client:process/client:c[1])`

> **Note:**
>
> Below the DOM, there is no `c` node with its `attr1` having `eng` as a value.

$OutputVarNameOnEB before assign	Result
<processResponse xmlns="http:// xmlns.oracle.com/SOA/ ComplexAssignment/BPELProcess1"> <a/> <x/> <c attr1="Old Value" attr2="Old Value"> <x>Old Value</x> <y>Old Value</y> <z>Old Value</z> </c> <d/> <e/> </processResponse>	This assignment fails at runtime with an exception.

- Runtime Error Assignment Example: `($OutputVarNameOnEB/client:processResponse/client:c[ns05:aCustomFunction(.)]).assign($InputVarNameOnEB/client:process/client:c[1])`
- Details: At runtime, the `ns05:aCustomFunction(.)` may return `false` or a number. Using the following DOM:
  - If `ns05:aCustomFunction(.)` returns `1`, the assignment occurs.
  - If `ns05:aCustomFunction(.)` returns `2`, this is the same as use case Append to an Indexed TO - The TO Element Has a Predicate for an APPEND Operation - Not Supported.
  - If `ns05:aCustomFunction(.)` returns false, a runtime error occurs.

$OutputVarNameOnEB before assign	Result
<processResponse xmlns="http:// xmlns.oracle.com/SOA/ ComplexAssignment/BPELProcess1"> <a/> <x/> <c attr1="Old Value" attr2="Old Value"> <x>Old Value</x> <y>Old Value</y> <z>Old Value</z> </c> <d/> <e/> </processResponse>	This assignment fails at runtime with an exception.

Extended Data Types

The following extended data types use cases and their expected results are described.

Schema Validation when <xsd:any> or xsd:anyType are in the location path - Use Case - ASSIGN - The "FROM" and/or "TO"

- Expected Result: You cannot type the location path steps after the `xsd:anyType` (or `<xsd:any>`'s parent). The location path is accepted, as long as the syntax is valid.
- Example Details: Populating the `AnyType` Element with complex content.

```
($v1/globalElement/element3AnyType).ASSIGN($list/ns1:items/
ns1:item_Undoulded)
```

$v1 before assign	$list content	outputVariable expected result
<globalElement xmlns="http://jorge.com/AssignAnyType2"> <element1 lang="esperanto"/> <element4>2017-09-27T14:31:00</element4> </globalElement>example	<items xmlns="http://xmlns.XYZ.com"> <item_Unboulded> <name>Music</name> <price>98.99</price> </item_Unboulded> <item_Unboulded> <name>Fighter</name> <price>300.99</price> </item_Unboulded> <item_Unboulded> <name>Lover</name> <price>20.99</price> </item_Unboulded> </items>example	<globalElement xmlns="http://jorge.com/AssignAnyType2"> <element1 lang="esperanto"/> <element3AnyType> <item_Unboulded xmlns="http://xmlns.XYZ.com"> <name>Music</name> <price>98.99</price> </item_Unboulded> <item_Unboulded xmlns="http://xmlns.XYZ.com"> <name>Fighter</name> <price>300.99</price> </item_Unboulded> <item_Unboulded xmlns="http://xmlns.XYZ.com"> <name>Lover</name> <price>20.99</price> </item_Unboulded> </element3AnyType> <element4>2017-09-27T14:31:00</element4> </globalElement>

- Example Details: Referencing data within the `any` Type.
  If you now want to reference the price of `item_Unboulded` 2:

```
($vFigtherPrice).ASSIGN($v1/ns0:globalElement/ns0:element3AnyType/
ns1:item_Unboulded[2]/ns1:price)
```

– `$vFighterPeice resultExample`:

```
300.99
```

If you now want to assign the name of element 2:

```
($v1/ns0:globalElement/ns0:element3AnyType/ns1:item_Unboulded[2]/
ns1:name).ASSIGN("Me Ma Mu")
```

– `outputVariable` expected result:

```
<!-- note the instance 2 name is changed from fighter to Me Ma Mu
<globalElement xmlns="http://jorge.com/AssignAnyType2">
    <element1 lang="esperanto"/>
    <element3AnyType>
        <price xmlns="http://jorge.com/price">300</price>
    </element3AnyType>
    <element3AnyType>
        <item_Unboulded xmlns="http://xmlns.XYZ.com">
            <name>Music</name>
            <price>98.99</price>
        </item_Unboulded>
        <item_Unboulded xmlns="http://xmlns.XYZ.com">
            <name>Me Ma Mu</name>
            <price>300.99</price>
        </item_Unboulded>
        <item_Unboulded xmlns="http://xmlns.XYZ.com">
            <name>Lover</name>
            <price>20.99</price>
        </item_Unboulded>
    </element3AnyType>
    <element4>2017-09-27T14:31:00</element4>
</globalElement>
```

Use Cases for <xsd:anyType>, <xsd:any>, and xsd:Any Defined in the Schema

The following XSD use cases and their expected results are described.

- For `<xsd:anyType>` and `<xsd:any>`:

```
Error rendering macro 'excerpt-include'
No link could be created for 'Expected Result - xsd:anyType and xsd:any'.
```

- For `xsd:Any` defined in the schema:

```
Error rendering macro 'excerpt-include'
No link could be created for 'Expected Result - 2 XSD:ANY'.
```