Oracle® Cloud
Using Integrations in Oracle Integration
## Contents

### Preface

<table>
<thead>
<tr>
<th>Audience</th>
<th>x</th>
</tr>
</thead>
<tbody>
<tr>
<td>Documentation Accessibility</td>
<td>x</td>
</tr>
<tr>
<td>Related Resources</td>
<td>x</td>
</tr>
<tr>
<td>Conventions</td>
<td>xi</td>
</tr>
</tbody>
</table>

### 1 Get Started with Integrations

<table>
<thead>
<tr>
<th>Run the Sample Integrations</th>
<th>1-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run the Echo Sample</td>
<td>1-3</td>
</tr>
<tr>
<td>Run the Hello World Sample</td>
<td>1-5</td>
</tr>
<tr>
<td>Run the Hello World Invoke Sample</td>
<td>1-9</td>
</tr>
<tr>
<td>Run the Hello World Data Map Invoke Sample</td>
<td>1-13</td>
</tr>
<tr>
<td>Run the File Transfer Sample</td>
<td>1-17</td>
</tr>
<tr>
<td>Run the Incident Details from Service Cloud Sample</td>
<td>1-21</td>
</tr>
<tr>
<td>Load Newer Versions of the Sample Integrations</td>
<td>1-25</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>About Integrations</th>
<th>1-26</th>
</tr>
</thead>
<tbody>
<tr>
<td>About Integrations Concepts</td>
<td>1-27</td>
</tr>
<tr>
<td>About Connections</td>
<td>1-27</td>
</tr>
<tr>
<td>Connection Creation</td>
<td>1-27</td>
</tr>
<tr>
<td>Related Topics</td>
<td>1-30</td>
</tr>
<tr>
<td>About Oracle Integration Integrations</td>
<td>1-30</td>
</tr>
<tr>
<td>About Mappings</td>
<td>1-31</td>
</tr>
<tr>
<td>Map Request Data Between Applications</td>
<td>1-33</td>
</tr>
<tr>
<td>Map Response Data Between Applications</td>
<td>1-33</td>
</tr>
<tr>
<td>About Mapping Multiple Sources to a Target</td>
<td>1-34</td>
</tr>
<tr>
<td>About Oracle Integration Enrichments</td>
<td>1-35</td>
</tr>
<tr>
<td>About Oracle Integration Lookups</td>
<td>1-35</td>
</tr>
<tr>
<td>About Oracle Integration Packages</td>
<td>1-36</td>
</tr>
<tr>
<td>About Connectivity Agents and Integrations Between On-Premises Applications and Oracle Integration</td>
<td>1-37</td>
</tr>
<tr>
<td>About the Connectivity Agent Framework</td>
<td>1-37</td>
</tr>
<tr>
<td>Workflow for Using the Connectivity Agent</td>
<td>1-39</td>
</tr>
</tbody>
</table>
Create Connections

Inbound Endpoints and Transport Layer Security Server Version 1.2 Support
Create a Connection
Add a Contact Email
Configure Connection Properties
Configure Connection Security
Test the Connection
Edit a Connection
Clone a Connection
Delete a Connection
Unlock a Connection
Refresh Integration Metadata
Manage Security Certificates
  Upload an SSL Certificate
  Update or Delete an SSL Certificate

Create Integrations

Understand Integration Patterns
Common Integration Pattern Pitfalls and Design Best Practices
  Avoid Common Integration Pattern Pitfalls
  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
Understand Trigger and Invoke Connections
Create Integrations
Create Application-Driven Orchestrated Integrations
  Create an Orchestrated Integration
  Add Actions, Connections, and Artifacts Through an Inline Menu
  Define Ad-Hoc Mappings
  Import a Map File into an Orchestrated Integration
  Add Actions to an Orchestrated Integration
Loop over Repeating Elements with a For-Each Action 3-24
Route Expressions with Switch Branches 3-27
Manage a Group of Actions and Fault Handlers with a Scope Action 3-28
Assign Values to Scalar Variables in an Assign Action 3-34
Loop Over Actions or Invoke Connections While a Condition is Satisfied with a While Action 3-38
Send Notification Emails During Stages of the Integration with a Notification Action 3-40
Delay Integration Execution for a Specified Time Period with a Wait Action 3-42
Add Global Fault Handling to Orchestrated Integrations 3-43
Catch Faults with a Re-throw Fault Action 3-47
Throw Faults with a Throw New Fault Action 3-49
Process Files in Scheduled Integrations with a Stage File Action 3-51
Log Messages with a Logger Action 3-62
Add a JavaScript Action 3-63
Add Placeholder Notes with a Note Action 3-64
Use XPath Axis and Wildcard Expressions in the Expression Builder 3-65
Use Lookups in Variable Assignments 3-66
Define Fault Aggregation in Parallel Processing Scenarios 3-68
Assign Business Identifiers for Tracking Fields 3-70
Display Errors and Warnings in an Integration 3-70
Define Inbound Triggers and Outbound Invokes 3-71
Create Basic Routing Integrations 3-74
Create a Basic Routing Integration 3-74
Add a Trigger (Source) Connection 3-74
Add an Invoke (Target) Connection 3-75
Add Request and Response Enrichments 3-76
Delete Request and Response Enrichments 3-79
Create Routing Paths for Two Different Invoke Endpoints in Integrations 3-79
Create Routing Expression Logic in Both Expression Mode and Condition Mode 3-84
Delete Routing Paths 3-91
Create Scheduled Integrations 3-92
Create a Scheduled Integration 3-93
Create a Schedule 3-93
Create Parameters in Scheduled Integrations 3-97
Override Schedule Parameter Values 3-99
Start and Pause an Integration Schedule 3-100
View Past and Present Integration Runs 3-101
View Future Runs 3-102
Edit an Integration Schedule 3-103
Delete an Integration Schedule 3-104
Unlock an Integration Schedule 3-104
Delete Runs 3-104
Monitor Integration Runs 3-105
Create Ad-Hoc Integration Runs Using the REST API 3-106
Activate Older, Imported Scheduled Integrations Using Operations Other Than Read Multiple Files 3-107
Create Integrations to Publish and Subscribe to Oracle Integration 3-108
Create an Integration to Publish Messages to Oracle Integration 3-108
Create an Integration to Subscribe to Oracle Integration 3-109
Assign Business Identifiers for Tracking Fields in Messages 3-111
Assign Business Identifiers 3-111
Delete Business Identifiers 3-113
Import a Prebuilt Integration 3-113
Import and Export Integrations 3-113
Export an Integration 3-114
Import an Integration 3-115
Regenerate a WSDL File for Integrations 3-115

4 Map Data and Create Lookups

Map Data 4-1
Access the Mapper 4-1
Modify Mappings 4-2
Delete All Mappings 4-3
Map Faults 4-3
Encode and Decode File Attachment Content 4-4
Add Customized Mappings to Prebuilt Integrations 4-6
Remove Customized Mappings from Prebuilt Integrations 4-8
Import Map Files 4-9
Import a Map File into Oracle JDeveloper 4-9
Import a Map File into Oracle Integration 4-11
Manage Lookups 4-12
Create a Lookup 4-12
Add Adapters or Domain Names to a Lookup 4-12
Clone a Lookup 4-13
Delete a Lookup 4-14
Update Lookups in Active Integrations 4-14
Export a Lookup 4-14
Import a Lookup 4-14
Create the lookupValue Function 4-15
Access the Build Lookup Function Wizard 4-15
Select the Lookup Table 4-16
Select the Source and Target Columns 4-16
### Manage the Agent Group and the On-Premises Connectivity Agent

- Create an Agent Group 5-1
- Download and Run the Connectivity Agent Installer 5-2
  - System Requirements 5-2
  - Restart the Agent 5-3
- Agent Download and Installation 5-3
  - Download and Install the Agent 5-4
  - Install a Certificate on the Agent Host 5-6
- Use the Agent in High Availability Environments 5-7
- Run the Connectivity Agent Installer as a Background Process 5-8
- Create a Connection with an Agent Group 5-9
- Upgrade the Connectivity Agent Automatically 5-9
- Back Up and Recover the Connectivity Agent 5-10
- Delete an Agent Group 5-11
- Agent Behavior in a Decommissioned Instance or HTTP 404/401 Error Response Codes 5-12
- Troubleshoot Connectivity Agent Issues 5-12

### Integrate with Processes and Services

- Create an Integration that Invokes a Process 6-1
- Connect to Oracle Mobile Cloud Service 6-2
- Connect to Oracle API Platform Cloud Service 6-2

### Manage Integrations

- Activate an Integration 7-1
- Deactivate an Integration 7-4
- Manage Packages 7-5
  - View the Integrations in a Package 7-5
  - Import a Package 7-6
  - Export a Package 7-6
  - Update a Package 7-6
  - Delete a Package 7-7
- Modify an Integration 7-7
- Edit the Endpoint Information in an Integration 7-8
- Enable or Disable Tracing on Active Integrations 7-9
- View the Trigger, Invoke, and Enrichment Details of an Integration 7-10
Clone an Integration  7-11
Create a Draft of an Integration  7-11
Delete an Integration  7-12
Unlock an Integration  7-12
Cancel a Running Integration Instance  7-13
Download Generated Artifacts for an Integration  7-13
View the Contents of an Incomplete Integration  7-13
Edit an Integration with Incomplete Connections  7-14
Reactivate Integrations After Instance Upgrade to View the Latest Business Identifier Tracking Behavior  7-14
Filter the Display of Integrations By Type  7-15
Change the Time Zone  7-15
Use Libraries to Manage Functions for Integrations  7-15
  Register a JavaScript Library  7-16
  View Library Details  7-17
  Edit Library Details  7-17
  Determine Which Integrations Use a Library Function  7-18
  Delete a Library  7-18
  Download a Library File  7-18
  Use Library Functions in Orchestrations  7-19
Manage an Integration as an API with Oracle API Platform Cloud Service  7-19
  Create a New API in Oracle API Platform Cloud Service Using an Integration  7-19
  Group an Integration in an API in Oracle API Platform Cloud Service  7-20
  Ungroup an Integration from an API  7-21

8 Monitor Integrations During Runtime

Monitor Integrations  8-1
  View the Dashboard  8-1
    View the Activity Stream  8-3
  View System Health  8-4
  View Design-Time Metrics  8-6
  Monitor Integrations  8-7
  Monitor Agents  8-8
  Monitor Scheduled Integration Runs  8-9
Manage Errors  8-10
  View Specific Error Details  8-10
  Resubmit Failed Messages  8-13
  Discard Errors  8-14
  View the Status of Message Recovery  8-15
  View the Audit Trail of Failed Integration Instances  8-15
  View Business Identifiers in Failed Integration Instances  8-16
9 Adapter Configuration Reference

Configure Trigger Oracle Integration Messaging 9-1

10 Troubleshoot Oracle Integration

Conflict with Using Variables of String Data Type and Applying Mathematical Comparator Expressions 10-1
Reduce Network Calls that Slow Down Processes Caused By Writing Record By Record 10-1
403 Error when Accessing an API Protected using OAuth 10-2
Troubleshoot Integration Activations 10-2
Invoke Subprocess Integrations Using a Single REST API Connection 10-4
Troubleshoot Certificate Import Failures 10-5
Trigger Connection Cannot Receive an Empty Array in JSON if the Corresponding JSON Received Has an Empty JSON 10-6
Preface

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

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

Topics:
- Audience
- Documentation Accessibility
- Related Resources
- Conventions

Audience

This guide is intended for 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.

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:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>boldface</strong></td>
<td>Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.</td>
</tr>
<tr>
<td><em>italic</em></td>
<td>Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.</td>
</tr>
<tr>
<td><strong>monospace</strong></td>
<td>Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.</td>
</tr>
</tbody>
</table>
1 Get Started with Integrations

Review the following topics to learn about how Integrations in Oracle Integration work. These topics provide information about Integrations concepts and components to help you get started with creating your own integrations.

Topics:

• Run the Sample Integrations
• About Integrations
• About Integrations Concepts
• About Monitoring
• About Error Management
• About Business Identifiers for Tracking Fields in Messages
• Start Oracle Integration
• Navigate Oracle Integration
• Typical Workflow for Creating Integrations with Oracle Integration

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.

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.

<table>
<thead>
<tr>
<th>Sample</th>
<th>Complexity</th>
<th>This Sample Demonstrates How to ...</th>
<th>See ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Echo</td>
<td>None</td>
<td>Use simple logging with a REST Adapter as a trigger in a request-response orchestrated integration.</td>
<td>Run the Echo Sample</td>
</tr>
<tr>
<td>Hello World</td>
<td>None</td>
<td>Use simple log and email notification actions with a REST Adapter in an orchestrated integration.</td>
<td>Run the Hello World Sample</td>
</tr>
<tr>
<td>Sample</td>
<td>Complexity</td>
<td>This Sample Demonstrates How to ...</td>
<td>See ...</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>------------</td>
<td>-----------------------------------------------------------------------------------------------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>Hello World Invoke</td>
<td>Minimal</td>
<td>Call and invoke a simple REST web service in an orchestrated integration. Simple logging is also used.</td>
<td>Run the Hello World Invoke Sample</td>
</tr>
<tr>
<td>Hello World Data Map Invoke</td>
<td>Minimal</td>
<td>Call and invoke a simple REST web service using the Map Data integration pattern.</td>
<td>Run the Hello World Data Map Invoke Sample</td>
</tr>
<tr>
<td>File Transfer</td>
<td>Medium</td>
<td>Read an opaque file from a &quot;/&quot; directory and write the file to an &quot;/upload&quot; 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.</td>
<td>Run the File Transfer Sample</td>
</tr>
<tr>
<td>Incident details from Service Cloud</td>
<td>Medium</td>
<td>Get incident details from the Oracle Service Cloud for an incident ID and send the incident details to the caller as a response.</td>
<td>Run the Incident Details from Service Cloud Sample</td>
</tr>
</tbody>
</table>

**Note:**
The documentation for running these samples is also available when you click the Information icon for an activated integration and select How to run.
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 navigation pane, click Integrations.
2. In the row for the Echo sample, click the Activate icon, then click Activate when prompted.
3. Wait for the icon to turn green, indicating that it is activated.

How To Run
1. Enter the following URL in a browser. You can also copy this URL from the Information icon for this sample.

   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.
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 navigation pane, click **Integrations**, then click the < arrow next to **Designer**.
2. Click **Monitoring**, then click **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. From the **Integration Health** dropdown menu, you can view overall system health and design time metrics.
3. Click **Activity Stream** to view details about the invocation.
4. In the navigation pane, click **Tracking** and note that the integration is listed as completed.

5. In the navigation pane, click **Integrations** and note the status of the message processed.

How To View

1. Click the icon.
2. In the navigation pane, click **Integrations**.
3. Click the **Echo** integration.
   A read-only version of the integration is displayed for viewing. Because the integration is active, it cannot be edited.
4. 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 /
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 navigation pane, click Integrations.
2. In the row for the Hello World sample, click the Activate icon, then click Activate when prompted.
3. Wait for the icon to turn green, indicating that it is activated.

How To Run
1. Enter one of the following URLs in a browser. You can also access these URLs from the Information 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

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:

```json
{
    "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 OIC!

How To Monitor

1. In the navigation pane, click **Integrations**, then click the < arrow next to **Designer**.
2. Click **Monitoring**, then click **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. From the **Integration Health** dropdown menu, you can view overall system health and design time metrics.

3. Click **Activity Stream** to view details about the invocation.
4. In the navigation pane, click **Tracking** and note that the **Hello World** integration instance is listed as completed.

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

How To View

1. Click the icon.
2. In the navigation pane, click **Integrations**.

3. 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.

4. 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 Sample Hello World Invoke REST Adapter used in this sample.

1. In the navigation pane, click Integrations, then click Connections.
2. Click Sample Hello World Invoke.
3. Click Configure Connectivity to specify information to connect to the application/endpoint and process requests.
   a. For Connection Type, select Swagger Definition URL.
   b. For Connection URL, enter the following:
      
      https://hostname:port/ic/api/integration/v1/flows/rest/HELLO_WORLD_INVOKE/1.0/metadata/swagger

   c. Click OK.
   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 Sample Hello World Invoke was tested successfully.
6. Click Save, then click Close.

How To Activate

1. In the navigation pane, click Integrations.
2. In the row for the Hello World Invoke sample, click the Activate icon, then click Activate when prompted.

3. Wait for the icon to turn green, indicating that it is activated.

How To Run

Enter one of the following URLs in a browser. You can also access these URLs from the Information 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

What Results Do You See

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

```json
{
    "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 navigation pane, click Integrations, then click the < arrow next to Designer.
2. Click Monitoring, then click 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. From the Integration Health dropdown menu, you can view overall system health and design time metrics.
3. Click Activity Stream to view details about the invocation.
4. In the 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.
5. In the navigation pane, click **Integrations** and note that the messages were successfully received and processed without any errors.

![Image](image.jpg)

**How To View**

1. Click the **home** icon.
2. In the navigation pane, click **Integrations**.
3. 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.

4. 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.

---

**Note:**

The examples in this document are intended for demonstration purposes only and may not reflect the exact configuration and functionality required for production use. It is recommended to thoroughly test and validate any integration before deploying it to a production environment.
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 Data Map Invoke Sample

This sample demonstrates how to call and invoke a simple REST web service using the Map Data integration pattern. 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 Sample Hello World Invoke REST Adapter used in this sample.

1. In the navigation pane, click Integrations, then click Connections.
2. Click Sample Hello World Invoke.
3. Click Configure Connectivity to specify information to connect to the application/endpoint and process requests.
   a. For Connection Type, select Swagger Definition URL.
   b. For Connection URL, enter the following:
      
      https://hostname:port/ic/api/integration/v1/flows/rest/HELLO_WORLD/1.0/metadata/swagger
   c. Click OK.
   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 Sample Hello World Invoke was tested successfully.
6. Click Save, then click Close.

How To Activate

1. In the navigation pane, click Integrations.
2. In the row for the Hello World Data Map Invoke sample, click the Activate icon, then click Activate when prompted.
3. Wait for the icon to turn green, indicating that it is activated.

How To Run

1. 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

What Results Do You See

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

   ```json
   {
     "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:

   ```json
   {
     "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 navigation pane, click **Integrations**, then click the < arrow next to **Designer**.

2. Click **Monitoring**, then click **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. From the **Integration Health** dropdown menu, you can view overall system health and design time metrics.

3. Click **Activity Stream** to view details about the invocation.

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

How To View

1. Click the icon.

2. In the navigation pane, click **Integrations**.

3. 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.

4. 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**

- Create Integrations
- 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 navigation pane, click **Integrations**.
2. In the row for the File Transfer sample, click the **Activate** icon, then click **Activate** when prompted.
3. Wait for the icon to turn green, indicating that it 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 navigation pane, click Integrations, then click Connections.
2. Click Sample FTP connection.
3. Click Configure Connectivity 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.
   c. Click OK.
   b. For Username, enter anonymous.
   c. Enter the same password twice. You can use any password.
   d. 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 Sample FTP connection was tested successfully.

6. Click Save, then click Close.

**How To Run**

1. From the Actions menu for the integration, select one of the following options:

   - Select Submit Now to run the scheduled integration immediately:
     a. View the following message at the top of the page. You are ready to monitor the results of the file transfer.

     **Submit Now** request to run integration File Transfer sample (version_number) was submitted successfully with request id number.
• Select **Add Schedule** to schedule a time at which to run the integration:
  
The Run File Transfer sample page is displayed.
  
  a. Accept the default values for **Type (Basic)** and **Frequency (Only Once)**.
  
  b. For **This schedule runs**, click the time and select **Modify start date**.
  
  c. Use the **Calendar** icon to specify a soon-to-occur start time, and click **OK**.
  
  d. Click **Save**, then click **Close**.
  
The Schedule and Future Runs page is displayed.
  
  e. Click **Start Schedule**.
  
The time at which the integration is scheduled to run is displayed at the bottom of the page.
  
  f. When the scheduled time passes, click the **refresh** icon. The scheduled integration is no longer listed, indicating that it has run.
  
  g. Click **Close**.

**How To Monitor**

1. In the navigation pane, click **Integrations**, then click the < arrow next to **Designer**.
  
2. Click **Monitoring**, then click **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. From the **Integration Health** dropdown menu, you can view overall system health and design time metrics.
  
3. Click **Activity Stream** to view details about the scheduled integration. The log indicates the 1 KB file was uploaded to the `/upload` directory.
4. In the navigation pane, click **Tracking** and note that the **File Transfer sample** integration instance is listed as completed. You can see the filename and file size, which enables you to look for particular files in case of problems.

5. In the 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. Click the **Home** icon.
2. In the navigation pane, click **Integrations**.
3. 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.

4. 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 navigation pane, click Integrations.
2. In the row for the Incident Details from Service Cloud sample, click the Activate icon, then click Activate when prompted.
3. Wait for the icon to turn green, indicating that it 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_INCIDDETAIL_FROM_SERVICEL/1.0/incident/12

For example:

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

What Results Do You See

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

```json
{
   "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 navigation pane, click Integrations, then click the < arrow next to Designer.
2. Click Monitoring, then click 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. From the Integration Health dropdown menu, you can view overall system health and design time metrics.
3. Click Activity Stream to view details about the incident ID.
4. In the navigation pane, click Tracking and note that the Incident details from Service Cloud integration instance is listed as completed.
In the navigation pane, click **Integrations** and note that the message was successfully received and processed without any errors.

**How To View**

1. Click the icon.
2. In the navigation pane, click **Integrations**.
3. 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.

4. 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

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 navigation pane, click **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 navigation pane, click **Integrations > Packages** to see the package names associated with each sample integration.

---

### 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 Engagement Cloud Adapter, Oracle Service Cloud, 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
• About Oracle Integration Integrations
• About Mappings
• About Oracle Integration Enrichments
• About Oracle Integration Lookups
• About Oracle Integration Packages
• About Connectivity Agents and Integrations Between On-Premises Applications and Oracle Integration

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 Engagement Cloud Adapter, Oracle Eloqua Cloud, 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.

<table>
<thead>
<tr>
<th>Adapter</th>
<th>For Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adobe eSign Adapter</td>
<td>Using the Adobe eSign Adapter</td>
</tr>
<tr>
<td>Concur Adapter</td>
<td>Using the Concur Adapter</td>
</tr>
<tr>
<td>DB2 Adapter</td>
<td>Using the DB2 Adapter</td>
</tr>
<tr>
<td>DocuSign Adapter</td>
<td>Using the DocuSign Adapter</td>
</tr>
<tr>
<td>Eventbrite Adapter</td>
<td>Using the Eventbrite Adapter</td>
</tr>
<tr>
<td>Evernote Adapter</td>
<td>Using the Evernote Adapter</td>
</tr>
<tr>
<td>Facebook Adapter</td>
<td>Using the Facebook Adapter</td>
</tr>
<tr>
<td>File Adapter</td>
<td>Using the File Adapter</td>
</tr>
<tr>
<td>FTP Adapter</td>
<td>Using the FTP Adapter</td>
</tr>
<tr>
<td>Gmail Adapter</td>
<td>Using the Gmail Adapter</td>
</tr>
<tr>
<td>Adapter</td>
<td>For Information</td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>------------------------------------------------------</td>
</tr>
<tr>
<td>Google Calendar Adapter</td>
<td>Using the Google Calendar Adapter</td>
</tr>
<tr>
<td>Google Task Adapter</td>
<td>Using the Google Task Adapter</td>
</tr>
<tr>
<td>LinkedIn Adapter</td>
<td>Using the LinkedIn Adapter</td>
</tr>
<tr>
<td>Microsoft Calendar Adapter</td>
<td>Using the Microsoft Calendar Adapter</td>
</tr>
<tr>
<td>Microsoft Contact Adapter</td>
<td>Using the Microsoft Contact Adapter</td>
</tr>
<tr>
<td>Microsoft Email Adapter</td>
<td>Using the Microsoft Email Adapter</td>
</tr>
<tr>
<td>Microsoft SQL Server Adapter</td>
<td>Using the Microsoft SQL Server Adapter</td>
</tr>
<tr>
<td>MailChimp Adapter</td>
<td>Using the MailChimp Adapter</td>
</tr>
<tr>
<td>MySQL Adapter</td>
<td>Using the MySQL Adapter</td>
</tr>
<tr>
<td>Oracle Advanced Queuing (AQ) Adapter</td>
<td>Using the Oracle Advanced Queuing (AQ) Adapter</td>
</tr>
<tr>
<td>Oracle Commerce Cloud Adapter</td>
<td>Using the Oracle Commerce Cloud Adapter</td>
</tr>
<tr>
<td>Oracle CPQ Cloud Adapter</td>
<td>Using the Oracle CPQ Cloud Adapter</td>
</tr>
<tr>
<td>Oracle Database Adapter</td>
<td>Using the Oracle Database Adapter</td>
</tr>
<tr>
<td>Oracle Database Cloud Service Adapter</td>
<td>Using the Oracle Database Cloud Service Adapter</td>
</tr>
<tr>
<td>Oracle Enterprise Performance Management Cloud Adapter</td>
<td>Using the Oracle Enterprise Performance Management Cloud Adapter</td>
</tr>
<tr>
<td>Oracle E-Business Suite Adapter</td>
<td>Using the Oracle E-Business Suite Adapter</td>
</tr>
<tr>
<td>Oracle Eloqua Cloud Adapter</td>
<td>Using the Oracle Eloqua Cloud Adapter</td>
</tr>
<tr>
<td>Oracle ERP Cloud Adapter</td>
<td>Using the Oracle ERP Cloud Adapter</td>
</tr>
<tr>
<td>Oracle Field Service Adapter</td>
<td>Using the Oracle Field Service Adapter</td>
</tr>
<tr>
<td>Oracle HCM Cloud Adapter</td>
<td>Using the Oracle HCM Cloud Adapter</td>
</tr>
<tr>
<td>Oracle JD Edwards EnterpriseOne Adapter</td>
<td>Using the Oracle JD Edwards EnterpriseOne Adapter</td>
</tr>
<tr>
<td>Oracle Logistics Adapter</td>
<td>Using the Oracle Logistics Adapter</td>
</tr>
<tr>
<td>Oracle Messaging Cloud Service Adapter</td>
<td>Using the Oracle Messaging Cloud Service Adapter</td>
</tr>
<tr>
<td>Oracle Monetization Cloud Adapter</td>
<td>Using the Oracle Monetization Cloud Adapter</td>
</tr>
<tr>
<td>Oracle NetSuite Adapter</td>
<td>Using the Oracle NetSuite Adapter</td>
</tr>
<tr>
<td>Oracle Policy Automation Adapter</td>
<td>Using the Oracle Policy Automation Adapter</td>
</tr>
<tr>
<td>Oracle Responsys Adapter</td>
<td>Using the Oracle Responsys Adapter</td>
</tr>
<tr>
<td>Oracle Engagement Cloud Adapter</td>
<td>Using the Oracle Engagement Cloud Adapter</td>
</tr>
<tr>
<td>Oracle Service Cloud (RightNow) Adapter</td>
<td>Using the Oracle Service Cloud (RightNow) Adapter</td>
</tr>
<tr>
<td>Oracle Siebel Adapter</td>
<td>Using the Oracle Siebel Adapter</td>
</tr>
<tr>
<td>Oracle Talent Acquisition Cloud (Taleo EE) Adapter</td>
<td>Using the Oracle Talent Acquisition Cloud (Taleo EE) Adapter</td>
</tr>
<tr>
<td>Oracle WebLogic JMS Adapter</td>
<td>Using the Oracle WebLogic JMS Adapter</td>
</tr>
<tr>
<td>Oracle Utilities Adapter</td>
<td>Using the Oracle Utilities Adapter</td>
</tr>
<tr>
<td>REST Adapter</td>
<td>Using the REST Adapter</td>
</tr>
<tr>
<td>Salesforce Adapter</td>
<td>Using the Salesforce Adapter</td>
</tr>
<tr>
<td>SAP Adapter</td>
<td>Using the SAP Adapter</td>
</tr>
<tr>
<td>SAP Ariba Adapter</td>
<td>Using the SAP Ariba Adapter</td>
</tr>
</tbody>
</table>
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 Engagement Cloud 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 ICS** option in the Create Integration — Select a Pattern 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 Engagement Cloud 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 ICS** option in the Create Integration — Select a Pattern 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 Oracle Integration Integrations

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.

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.

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 place your cursor over elements and attributes to display 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 Engagement Cloud Adapter. The Oracle Engagement Cloud 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.

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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. You can view the XSLT code being created as you design your mappings. Once you complete designing your mappings, you can test them by entering sample content of the message to process in the mapping tester. You can select the following options:  
  - Select flow control mode, which enables you to add XSLT statements to your mapping, such as choose, for each, if, otherwise, when, copy-of, text, and value-of. A target element must already be created for you to drag an XSLT statement onto a target element node. Select this option, then select in the mapper toolbar to show the XSLT statements.  
  - 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. You can filter the display of element nodes, error messages, and warnings in the source or target data structures. |
<p>| Code | |
| Test | |
| View | |
| Filter | |</p>
<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Undo" /></td>
<td>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.</td>
</tr>
<tr>
<td><img src="image" alt="Redo" /></td>
<td>You can redo the action that was undone.</td>
</tr>
<tr>
<td><img src="image" alt="Maximize" /></td>
<td>You can maximize the size of the mapper. This is useful when working with large schemas.</td>
</tr>
<tr>
<td><img src="image" alt="Add Functions" /></td>
<td>You can add functions, operators, and XSLT expressions to your mappings.</td>
</tr>
</tbody>
</table>

**Video**

Map Request Data Between Applications

Once you create an integration and have the trigger (source) and invoke (target) in place, you can define how data is mapped between the two data structures.

The mapper appears with the source data structure on the left and the target data structure on the right:

1. Map request data between the source data structure and target data structure.
2. Click **Close**.

When returning from the mapper, the map icon changes color to indicate it is complete. Once you create a mapping in an integration, you can return to the mapping and make any necessary changes to how you mapped your data.

See Mapping Data of *Using the Oracle Mapper*.

Map Response Data Between Applications

If your integration pattern contains a response, you can map the response.

1. Map response data between the source data structure and target data structure.
2. Click **Close**.

When returning from the mapper, the map icon changes color to indicate it is complete.

Once you create a mapping in an integration, you can return to the mapping and make any necessary changes to how you mapped your data.

See Mapping Data of *Using the Oracle Mapper*.
About Mapping Multiple Sources to a Target

When mapping data between source and target data structures, some integration scenarios enable you to map the fields of multiple source structures to the fields of a single target structure.

Integration scenarios that include multiple source structure capabilities include the following:

- Integrations in which message enrichment points have been added (for example, a request message enrichment point, a response message enrichment point, or both points). For example, within the context of the following inbound trigger connection to outbound invoke connection, request mappings and request enrichment mappings are both defined.

![Diagram](image1)

Clicking the **Request Mapping** icon shows that there are two sources available for mapping in the **Source** section. The **process** structure is the primary source. The **$RequestEnrichmentApplicationObject** structure is the secondary source. Secondary sources are treated as variables and identified by the $ added to the front. The fields of both sources can be mapped to the fields of the target.

![Source](image2)

- Integration responses with a response mapping between a trigger connection and an invoke connection. For example, within the context of the invoke connection’s response back to the trigger connection, there are response mappings.

![Diagram](image3)
Clicking the **Response Mapping** icon shows the two sources available for mapping in the **Source** section. The **GetResponse** structure is the primary source. The **$SourceApplicationObject** structure is the secondary source (note the $). The fields of both sources can be mapped to the fields of the target.

See Mapping Multiple Sources to a Target of *Using the Oracle Mapper*.

### 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.

**Lookup Function**

Oracle Integration provides a `lookupValue` function that you can call in the mapper to specify when to reference a lookup table. Use this function to look up values at runtime based on information in incoming messages. This way, your integration knows how to map data coming in from one application to data being sent to another application.

![Mapping Components](image)

See Referencing Lookups of *Using the Oracle Mapper*.

**Video**

**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.
The on-premises connectivity agent enables you to create integrations and exchange messages between on-premises applications and Oracle Integration. Message payloads of up to 10 MB are supported through the use of compression. The on-premises connectivity agent provides multithreading support, which allows for multiple executors to perform downstream message processing.

This type of integration enables you to:

- 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 Engagement Cloud Adapter, Oracle CPQ, Oracle Service Cloud, and Salesforce.com

About the Connectivity Agent Framework

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:
  – DB2
  – File
  – Microsoft SQL Server
  – MySQL Database
  – Oracle Database
  – Oracle E-Business Suite
  – REST
  – SAP
  – Siebel
  – SOAP

• Inbound (trigger) adapters: The following adapters can be configured as trigger connections in an integration:
  – DB2
  – File
  – JMS
  – Microsoft SQL Server
  – MySQL Database
  – Oracle Database
  – Oracle E-Business Suite
  – SAP
  – Siebel

Workflow for Using the Connectivity Agent

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create a connectivity agent group.</td>
<td>Create an Agent Group</td>
</tr>
<tr>
<td>Task</td>
<td>Documentation</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Download and run the on-premises connectivity agent installer on your host.</td>
<td>Download and Run the Connectivity Agent Installer</td>
</tr>
<tr>
<td>During installation setup, you associate the on-premises connectivity agent with the agent group.</td>
<td></td>
</tr>
<tr>
<td>Create an adapter connection in Oracle Integration and associate the connection with the connectivity agent group.</td>
<td>Create Connections</td>
</tr>
<tr>
<td>Design an integration that uses this connection.</td>
<td>Create Integrations</td>
</tr>
<tr>
<td>Activate the integration.</td>
<td>Activate an Integration</td>
</tr>
</tbody>
</table>

### Video

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 audit trail 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 Tracking 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 following URL, where hostname is the name of the host you received in your email after provisioning Oracle Integration, and port_number is the port number on which the server is listening:
   
   https://hostname:port_number/ic

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

   Once you sign in, the Welcome 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

Oracle Integration provides multiple ways to access the different features you use to create and monitor your integrations.

Using the Overlays

Overlays are displayed the first time you log in to Oracle Integration. Overlays are provided on many of the main pages. Overlays guide you on how to perform tasks such as getting started, creating connections, creating integrations, and so on.

1. Click Got it! to dismiss the overlays.
2. To enable the overlays again, select > Guide Me at the top of the page.

Using the Navigation Pane

1. In the navigation pane, click Integrations.
This takes you to the Integrations page. The navigation pane shows the following options.

2. Click the < arrow next to Designer to access additional options:
The navigation pane provides you with access to all the development features.  

3. Click the appropriate design feature:

<table>
<thead>
<tr>
<th>If You Click...</th>
<th>Links to the Following Features are Displayed...</th>
</tr>
</thead>
</table>
| Designer        | • Integrations  
                | • Connections  
                | • Lookups  
                | • Packages  
                | • Agents  
                | • Adapters  
                | • Libraries |
| Monitoring      | • Dashboards  
                | • Integrations  
                | • Agents  
                | • Tracking  
                | • Runs  
                | • Errors |
| Settings        | • Users (if using on-premises Oracle Integration)  
                | • Certificates  
                | • Notifications  
                | • Database  
                | • Logging Levels  
                | • Recommendations  
                | • API Platform  
                | • Tracing |

**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.
<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
<th>More Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>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.</td>
<td>Create a Connection</td>
</tr>
<tr>
<td>2</td>
<td>(Optional) Create lookups that map the different values used by those applications to identify the same type of object (such as gender codes or country codes).</td>
<td>Create a Lookup</td>
</tr>
</tbody>
</table>
| 3    | 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. Add a Source Connection  
3. Add a Target Connection  
4. Assign Business Identifiers  
5. Map Data in *Using the Oracle Mapper* |
| 4    | Activate the integration. | Activate an Integration |
| 5    | Monitor the integration on the dashboard. | View the Dashboard |
| 6    | Track payload fields in messages during runtime. | Assign Business Identifiers and Track Business Identifiers in Integrations During Runtime |
| 7    | Manage errors at the integration level, connection level, or specific integration instance level. | Manage Errors |
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
- Create a Connection
- Add a Contact Email
- Configure Connection Properties
- Configure Connection Security
- Test the Connection
- Edit a Connection
- Clone a Connection
- Delete a Connection
- Unlock a Connection
- Refresh Integration Metadata
- Manage Security Certificates

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.

Create a Connection

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

1. In the navigation pane, click Integrations, then click Connections.
2. Click Create.
Note:

You can also create a connection in the integration canvas of:

• An orchestrated integration (See Define Inbound Triggers and Outbound Invokes.)
• A basic routing integration (See Add a Trigger (Source) Connection.)

The Create Connection — Select Adapter dialog is displayed.

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

The Create New Connection dialog is displayed.

4. Enter the information to describe the connection.
   • Enter a meaningful name to help others find your connection when they begin to create their own integrations. The name you enter is automatically added in capital letters to the Identifier field. If you modify the identifier name, do not include a blank space (for example, Sales Opportunity).
   • Select the role (direction) in which to use this connection (trigger, invoke, or both). Only the roles supported by this adapter are displayed for selection. When you select a role, only the connection properties and security policies appropriate to that role are displayed on the Connections page. If you select an adapter that supports both invoke and trigger, but select only one of those roles, then try to drag the adapter into the section you did not select, you receive an error (for example, configure an Oracle Service Cloud (RightNow) Adapter as only an invoke, but drag the adapter to the trigger section).
   • Enter an optional description of the connection.
5. Click **Create**.
   
   Your connection is created and you are now ready to configure connection details, such as email contact, connection properties, security policies, connection login credentials, and (for certain connections) agent group.

### Add a Contact Email

You can add an optional contact email address for notifications.

1. In the **Email Address** field, enter an optional email address. You do *not* receive automatic notifications at this address.
2. In the upper right corner, click **Save**.

### Configure Connection Properties

Enter connection information so your application can process requests.

1. Click **Configure Connectivity**.
   
   The Connection Properties dialog is displayed.
2. See the following sections for information about specifying connection properties.

<table>
<thead>
<tr>
<th>Adapter</th>
<th>For Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adobe eSign Adapter</td>
<td>Using the Adobe eSign Adapter</td>
</tr>
<tr>
<td>Concur Adapter</td>
<td>Using the Concur Adapter</td>
</tr>
<tr>
<td>DB2 Adapter</td>
<td>Using the DB2 Adapter</td>
</tr>
<tr>
<td>DocuSign Adapter</td>
<td>Using the DocuSign Adapter</td>
</tr>
<tr>
<td>Eventbrite Adapter</td>
<td>Using the Eventbrite Adapter</td>
</tr>
<tr>
<td>Evernote Adapter</td>
<td>Using the Evernote Adapter</td>
</tr>
<tr>
<td>Facebook Adapter</td>
<td>Using the Facebook Adapter</td>
</tr>
<tr>
<td>File Adapter</td>
<td>Using the File Adapter</td>
</tr>
<tr>
<td>FTP Adapter</td>
<td>Using the FTP Adapter</td>
</tr>
<tr>
<td>Gmail Adapter</td>
<td>Using the Gmail Adapter</td>
</tr>
<tr>
<td>Google Calendar Adapter</td>
<td>Using the Google Calendar Adapter</td>
</tr>
<tr>
<td>Google Task Adapter</td>
<td>Using the Google Task Adapter</td>
</tr>
<tr>
<td>LinkedIn Adapter</td>
<td>Using the LinkedIn Adapter</td>
</tr>
<tr>
<td>Microsoft Calendar Adapter</td>
<td>Using the Microsoft Calendar Adapter</td>
</tr>
<tr>
<td>Microsoft Contact Adapter</td>
<td>Using the Microsoft Contact Adapter</td>
</tr>
<tr>
<td>Microsoft Email Adapter</td>
<td>Using the Microsoft Email Adapter</td>
</tr>
<tr>
<td>Microsoft SQL Server Adapter</td>
<td>Using the Microsoft SQL Server Adapter</td>
</tr>
<tr>
<td>MailChimp Adapter</td>
<td>Using the MailChimp Adapter</td>
</tr>
<tr>
<td>MySQL Adapter</td>
<td>Using the MySQL Adapter</td>
</tr>
<tr>
<td>Oracle Advanced Queuing (AQ) Adapter</td>
<td>Using the Oracle Advanced Queuing (AQ) Adapter</td>
</tr>
<tr>
<td>Oracle Commerce Cloud Adapter</td>
<td>Using the Oracle Commerce Cloud Adapter</td>
</tr>
<tr>
<td>Oracle CPQ Cloud Adapter</td>
<td>Using the Oracle CPQ Cloud Adapter</td>
</tr>
<tr>
<td>Oracle Database Adapter</td>
<td>Using the Oracle Database Adapter</td>
</tr>
<tr>
<td>Oracle Database Cloud Service Adapter</td>
<td>Using the Oracle Database Cloud Service Adapter</td>
</tr>
</tbody>
</table>
### Configure Connection Security

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

3. **Click OK.**

You are now ready to configure connection security.
1. Click **Configure Security**.

2. See the following sections for information about selecting the security policy and entering your login credentials.

<table>
<thead>
<tr>
<th>Adapter</th>
<th>For Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adobe eSign Adapter</td>
<td>Using the Adobe eSign Adapter</td>
</tr>
<tr>
<td>Concur Adapter</td>
<td>Using the Concur Adapter</td>
</tr>
<tr>
<td>DB2 Adapter</td>
<td>Using the DB2 Adapter</td>
</tr>
<tr>
<td>DocuSign Adapter</td>
<td>Using the DocuSign Adapter</td>
</tr>
<tr>
<td>Eventbrite Adapter</td>
<td>Using the Eventbrite Adapter</td>
</tr>
<tr>
<td>Evernote Adapter</td>
<td>Using the Evernote Adapter</td>
</tr>
<tr>
<td>Facebook Adapter</td>
<td>Using the Facebook Adapter</td>
</tr>
<tr>
<td>File Adapter</td>
<td>NA</td>
</tr>
<tr>
<td>FTP Adapter</td>
<td>Using the FTP Adapter</td>
</tr>
<tr>
<td>Gmail Adapter</td>
<td>Using the Gmail Adapter</td>
</tr>
<tr>
<td>Google Calendar Adapter</td>
<td>Using the Google Calendar Adapter</td>
</tr>
<tr>
<td>Google Task Adapter</td>
<td>Using the Google Task Adapter</td>
</tr>
<tr>
<td>Oracle JD Edwards EnterpriseOne Adapter</td>
<td>Using the JD Edwards EnterpriseOne Adapter</td>
</tr>
<tr>
<td>Oracle WebLogic JMS Adapter</td>
<td>Using the JMS Adapter</td>
</tr>
<tr>
<td>LinkedIn Adapter</td>
<td>Using the LinkedIn Adapter</td>
</tr>
<tr>
<td>Microsoft Calendar Adapter</td>
<td>Using the Microsoft Calendar Adapter</td>
</tr>
<tr>
<td>Microsoft Contact Adapter</td>
<td>Using the Microsoft Contact Adapter</td>
</tr>
<tr>
<td>Microsoft Email Adapter</td>
<td>Using the Microsoft Email Adapter</td>
</tr>
<tr>
<td>Microsoft SQL Server Adapter</td>
<td>Using the Microsoft SQL Server Adapter</td>
</tr>
<tr>
<td>MailChimp Adapter</td>
<td>Using the MailChimp Adapter</td>
</tr>
<tr>
<td>MySQL Adapter</td>
<td>Using the MySQL Adapter</td>
</tr>
<tr>
<td>Oracle Advanced Queuing (AQ) Adapter</td>
<td>Using the Oracle Advanced Queuing (AQ) Adapter</td>
</tr>
<tr>
<td>Oracle Commerce Cloud Adapter</td>
<td>Using the Oracle Commerce Cloud Adapter</td>
</tr>
<tr>
<td>Oracle CPQ Cloud Adapter</td>
<td>Using the Oracle CPQ Adapter</td>
</tr>
<tr>
<td>Oracle Database Adapter</td>
<td>Using the Oracle Database Adapter</td>
</tr>
<tr>
<td>Oracle Database Cloud Service Adapter</td>
<td>Using the Oracle Database Cloud Service Adapter</td>
</tr>
<tr>
<td>Oracle E-Business Suite Adapter</td>
<td>Using Oracle E-Business Suite Adapter</td>
</tr>
<tr>
<td>Oracle Eloqua Cloud Adapter</td>
<td>Using the Oracle Eloqua Cloud Adapter</td>
</tr>
<tr>
<td>Oracle Enterprise Performance Management Adapter</td>
<td>Using the Oracle Enterprise Performance Management Adapter</td>
</tr>
<tr>
<td>Oracle ERP Cloud Adapter</td>
<td>Using the Oracle ERP Cloud Adapter</td>
</tr>
<tr>
<td>Oracle Field Service Adapter</td>
<td>Using Oracle Field Service Adapter</td>
</tr>
<tr>
<td>Oracle HCM Cloud Adapter</td>
<td>Using the Oracle HCM Cloud Adapter</td>
</tr>
<tr>
<td>Oracle Logistics Adapter</td>
<td>Using the Oracle Logistics Adapter</td>
</tr>
<tr>
<td>Oracle Messaging Cloud Service Adapter</td>
<td>Using the Oracle Messaging Cloud Service Adapter</td>
</tr>
<tr>
<td>Oracle Monetization Cloud Adapter</td>
<td>Using the Oracle Monetization Cloud Adapter</td>
</tr>
<tr>
<td>Oracle NetSuite Adapter</td>
<td>Using the Oracle NetSuite Adapter</td>
</tr>
<tr>
<td>Oracle Policy Automation Adapter</td>
<td>Using the Oracle Policy Automation Adapter</td>
</tr>
</tbody>
</table>
3. Click OK.

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 is successfully configured.

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

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

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

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

4. When complete, click **Save**, then click **Close**.

---

<table>
<thead>
<tr>
<th>Adapter</th>
<th>For Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle Responsys Adapter</td>
<td>Using the Oracle Responsys Adapter</td>
</tr>
<tr>
<td>Oracle Engagement Cloud Adapter</td>
<td>Using the Oracle Engagement Cloud Adapter</td>
</tr>
<tr>
<td>Oracle Service Cloud (RightNow) Adapter</td>
<td>Using the Oracle Service Cloud (RightNow) Adapter</td>
</tr>
<tr>
<td>Oracle Siebel Adapter</td>
<td>Using the Oracle Siebel Adapter</td>
</tr>
<tr>
<td>Oracle Talent Acquisition Cloud (Taleo EE) Adapter</td>
<td>Using the Oracle Talent Acquisition Cloud (Taleo EE) Adapter</td>
</tr>
<tr>
<td>REST Adapter</td>
<td>Using the REST Adapter</td>
</tr>
<tr>
<td>Salesforce Adapter</td>
<td>Using the Salesforce Adapter</td>
</tr>
<tr>
<td>SAP Adapter</td>
<td>Using the SAP Adapter</td>
</tr>
<tr>
<td>SAP Ariba Adapter</td>
<td>Using the SAP Ariba Adapter</td>
</tr>
<tr>
<td>ServiceNow Adapter</td>
<td>Using the ServiceNow Adapter</td>
</tr>
<tr>
<td>SuccessFactors Adapter</td>
<td>Using the SuccessFactors Adapter</td>
</tr>
<tr>
<td>SOAP Adapter</td>
<td>Using the SOAP Adapter</td>
</tr>
<tr>
<td>SurveyMonkey Adapter</td>
<td>Using the SurveyMonkey Adapter</td>
</tr>
<tr>
<td>Trello Adapter</td>
<td>Using the Trello Adapter</td>
</tr>
<tr>
<td>Twilio Adapter</td>
<td>Using the Twilio Adapter</td>
</tr>
<tr>
<td>Twitter Adapter</td>
<td>Using the Twitter Adapter</td>
</tr>
<tr>
<td>Workday Adapter</td>
<td>Using the Workday Adapter</td>
</tr>
</tbody>
</table>
Edit a Connection

You can edit connection settings after creating a new connection.

1. In the navigation pane, click **Integrations**, then click **Connections**.
2. On the Connections page, search for the connection name.
3. Select **Edit** from the connection **Actions** menu or click the connection name.

![Actions icon]

**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 **Integrations**, then click **Connections**.
2. On the Connections page, search for the connection name.
3. Select **Clone** from the connection **Actions** menu.

   ![Actions](image)

   The Clone Connection dialog is displayed.

4. Enter the connection information.
5. Click **Clone**.
6. Click **Edit** 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 navigation pane, click **Integrations**, then click **Connections**.
2. On the Connections page, search for the connection name.
3. Click **Delete** from the connection **Actions** menu.

   ![Actions](image)

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

4. Click **Yes** 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 navigation pane, click **Integrations**, then click **Connections**.

3. For the connection that is locked, click the menu at the far right, then select **Unlock**.

---

**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:

1. In the navigation pane, click **Integrations**, then click **Connections**.

2. Locate the connection to refresh.

3. From the menu at the right, select **Refresh Metadata**.

A message is displayed indicating that the refresh was successful.

Metadata refresh for connection "connection_type" has been initiated successfully.
Manage Security Certificates

You can manage security certificates in Oracle Integration.

Topics:
• Upload an SSL Certificate
• Update or Delete an SSL Certificate

Upload an SSL Certificate

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

To upload an SSL certificate:

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

   All certificates currently uploaded to the trust store are displayed in the Certificates dialog. The **Filter By > Type** list displays the following details:
   • **Preinstalled**: Displays the certificates automatically installed in Oracle Integration. These certificates cannot be deleted.
   • **Uploaded**: Displays the certificates uploaded by individual users. These certificates can be deleted and updated.

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

3. Click **Upload** at the top of the page.
4. In the Upload Certificate dialog box, select the certificate type. Each certificate type enables Oracle Integration to connect with external services.
   • **Trust Certificate**: Use this option to upload a trust certificate.
     a. Enter a unique alias for the certificate.
     b. Click **Browse**, then select the trust file (for example, **.cer** or **.crt**) to upload.
   • **Message Protection Certificate**: Use this option to upload a keystore certificate with SAML token support. Create, read, update, and delete (CRUD) operations are supported on this type of certificate.
     a. Enter a unique alias for the certificate.
     b. Click **Browse**, then select the certificate file (**.cer** or **.crt**) to upload.
   • **Identity Certificate**: Use this option to upload a certificate for two-way SSL communication.
a. Click **Browse**, then select the keystore file (`.jks`) to upload.

b. Enter the password of the keystore being imported.

c. Enter the comma-separated list of aliases from the keystore being imported.

d. Enter the comma-separated list of passwords corresponding to key aliases.

e. If you want to display the passwords in clear text, select **Show Key Password(s)**. This enables you to ensure that you are correctly entering a list of keystore passwords.

5. Click **Upload**.

6. Click the certificate name to view details such as the subject of the certificate, the issuer of the certificate, the date the certificate was issued, and the date the certificate expires.

### 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. In the navigation pane, click **Integrations**, then click the `<` arrow next to **Designer**.

2. Click **Settings > Certificates**.

3. 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 **Filter By > Type > Preinstalled** or **Uploaded**.

   b. Search by entering a partial or complete certificate name in the **Search** field. To remove search or filter criteria, click the `x` icon in the banner.

4. At the far right of the certificate name, click the menu.

5. 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.

6. To delete the certificate, click **Delete**.

   • Click **Yes** when prompted to confirm your selection.
Create Integrations

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 add and remove request and response enrichment triggers. Click one of the following topics for more information.

Topics:
- Understand Integration Patterns
- Common Integration Pattern Pitfalls and Design Best Practices
- Understand Trigger and Invoke Connections
- Create Integrations
- Create Application-Driven Orchestrated Integrations
- Create Basic Routing Integrations
- Create Scheduled Integrations
- Create Integrations to Publish and Subscribe to Oracle Integration
- Assign Business Identifiers for Tracking Fields in Messages
- Import a Prebuilt Integration
- Import and Export Integrations
- Regenerate a WSDL File for Integrations

Understand Integration Patterns

You can select from several types of patterns when creating an integration in the Create Integration - Select a Style dialog.
<table>
<thead>
<tr>
<th>Pattern</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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 Engagement Cloud 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. |
| 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.  
  See Create a Basic Routing Integration. |
<table>
<thead>
<tr>
<th>Pattern</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Publish to OIC</td>
<td>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 pattern can be activated and deactivated independently of each other. See Create an Integration to Publish Messages to Oracle Integration.</td>
</tr>
<tr>
<td>Subscribe to OIC</td>
<td>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 pattern can be activated and deactivated independently of each other. See Create an Integration to Subscribe to Oracle Integration.</td>
</tr>
</tbody>
</table>

Common Integration Pattern Pitfalls and Design Best Practices

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

- Avoid Common Integration Pattern Pitfalls
- 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 Pattern Pitfalls

Designing integrations correctly from the start can save you a tremendous amount of rework. This section describes common integration pattern 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).

<table>
<thead>
<tr>
<th>Antipattern</th>
<th>Why an Antipattern?</th>
<th>Best Practice</th>
</tr>
</thead>
</table>
| 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

The scheduled integration looks for all files to process and loops over all to sequentially process until no files remain.

#### Why an Antipattern?

- 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.

#### Best Practice

- 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

Updating the IAR file externally and then importing it into Oracle Integration.

#### Why an Antipattern?

- This can lead to metadata inconsistency and validation failures.
- Activation failures may occur.

#### Best Practice

- 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.
<table>
<thead>
<tr>
<th>Antipattern</th>
<th>Why an Antipattern?</th>
<th>Best Practice</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reading the entire file in memory using the</td>
<td>• Consumes large amounts of memory and impacts other system processing.</td>
<td>• Download the file to the stage location using the download file option.</td>
</tr>
<tr>
<td>read file option and processing record by</td>
<td>• Sequential processing does not leverage built-in map reduction capabilities.</td>
<td></td>
</tr>
<tr>
<td>record.</td>
<td></td>
<td>• Use the read file with segments option.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

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

### 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 patterns internally use an asynchronous request response. Therefore, a synchronous integration using a scheduled orchestration is an anti-pattern.
• 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 timeout. 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 timeout. For instance, the default proxy on Oracle Public Cloud has a timeout value of 120 seconds. If the call is to an on-premises service behind a firewall, the proxy configured may also have its own timeout value.

In case timeouts are defined at multiple layers, the service invocation fails at the first timeout.

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 timeout 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 as business object, 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 connection.

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.

<table>
<thead>
<tr>
<th>Adapter</th>
<th>For Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2 Adapter</td>
<td>Using the DB2 Adapter</td>
</tr>
<tr>
<td>File Adapter</td>
<td>Using the File Adapter</td>
</tr>
<tr>
<td>FTP Adapter</td>
<td>Using the FTP Adapter</td>
</tr>
<tr>
<td>Oracle JD Edwards EnterpriseOne Adapter</td>
<td>Using the JD Edwards EnterpriseOne Adapter</td>
</tr>
<tr>
<td>MySQL Adapter</td>
<td>Using the MySQL Adapter</td>
</tr>
<tr>
<td>Oracle Advanced Queuing (AQ) Adapter</td>
<td>Using the Oracle Advanced Queuing (AQ) Adapter</td>
</tr>
<tr>
<td>Oracle Commerce Cloud Adapter</td>
<td>Using the Oracle Commerce Cloud Adapter</td>
</tr>
<tr>
<td>Oracle CPQ Cloud Adapter</td>
<td>Using the Oracle CPQ Adapter</td>
</tr>
<tr>
<td>Oracle Database Adapter</td>
<td>Using the Oracle Database Adapter</td>
</tr>
<tr>
<td>Oracle E-Business Suite Adapter</td>
<td>Using Oracle E-Business Suite Adapter</td>
</tr>
<tr>
<td>Oracle Eloqua Cloud Adapter</td>
<td>Using the Oracle Eloqua Cloud Adapter</td>
</tr>
<tr>
<td>Oracle ERP Cloud Adapter</td>
<td>Using the Oracle ERP Cloud Adapter</td>
</tr>
<tr>
<td>Oracle HCM Cloud Adapter</td>
<td>Using the Oracle HCM Cloud Adapter</td>
</tr>
<tr>
<td>Oracle Logistics Adapter</td>
<td>Oracle Logistics Cloud Adapter</td>
</tr>
<tr>
<td>Oracle Monetization Cloud Adapter</td>
<td>Using the Oracle Monetization Cloud Adapter</td>
</tr>
<tr>
<td>Oracle Messaging Cloud Service Adapter</td>
<td>Using the Oracle Messaging Cloud Service Adapter</td>
</tr>
<tr>
<td>Oracle Service Cloud (RightNow) Adapter</td>
<td>Using the Oracle Service Cloud (RightNow) Adapter</td>
</tr>
<tr>
<td>Oracle Engagement Cloud Adapter</td>
<td>Using the Oracle Engagement Cloud Adapter</td>
</tr>
<tr>
<td>Oracle Siebel Adapter</td>
<td>Using the Oracle Siebel Adapter</td>
</tr>
<tr>
<td>Oracle Utilities Adapter</td>
<td>Using the Oracle Utilities Adapter</td>
</tr>
<tr>
<td>REST Adapter</td>
<td>Using the REST Adapter</td>
</tr>
<tr>
<td>Salesforce Adapter</td>
<td>Using the Salesforce Adapter</td>
</tr>
<tr>
<td>SAP Adapter</td>
<td>Using the SAP Adapter</td>
</tr>
<tr>
<td>ServiceNow Adapter</td>
<td>Using the ServiceNow Adapter</td>
</tr>
<tr>
<td>SOAP Adapter</td>
<td>Using the SOAP Adapter</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Adapter</th>
<th>For Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adobe eSign Adapter</td>
<td>Using the Adobe eSign Adapter</td>
</tr>
<tr>
<td>Concur Adapter</td>
<td>Using the Concur Adapter</td>
</tr>
<tr>
<td>DB2 Adapter</td>
<td>Using the DB2 Adapter</td>
</tr>
<tr>
<td>DocuSign Adapter</td>
<td>Using the DocuSign Adapter</td>
</tr>
<tr>
<td>Eventbrite Adapter</td>
<td>Using the Eventbrite Adapter</td>
</tr>
<tr>
<td>Evernote Adapter</td>
<td>Using the Evernote Adapter</td>
</tr>
<tr>
<td>Facebook Adapter</td>
<td>Using the Facebook Adapter</td>
</tr>
<tr>
<td>File Adapter</td>
<td>Using the File Adapter</td>
</tr>
<tr>
<td>FTP Adapter</td>
<td>Using the FTP Adapter (Basic Map Integration)</td>
</tr>
<tr>
<td></td>
<td>Using the FTP Adapter (App Driven Integration)</td>
</tr>
<tr>
<td>Gmail Adapter</td>
<td>Using the Concur Adapter</td>
</tr>
<tr>
<td>Google Calendar Adapter</td>
<td>Using the Google Calendar Adapter</td>
</tr>
<tr>
<td>Google Task Adapter</td>
<td>Using the Google Task Adapter</td>
</tr>
<tr>
<td>Oracle WebLogic JMS Adapter</td>
<td>Using the JMS Adapter</td>
</tr>
<tr>
<td>LinkedIn Adapter</td>
<td>Using the LinkedIn Adapter</td>
</tr>
<tr>
<td>MailChimp Adapter</td>
<td>Using the MailChimp Adapter</td>
</tr>
<tr>
<td>Microsoft Calendar Adapter</td>
<td>Using the Microsoft Calendar Adapter</td>
</tr>
<tr>
<td>Microsoft Contact Adapter</td>
<td>Using the Microsoft Contact Adapter</td>
</tr>
<tr>
<td>Microsoft Email Adapter</td>
<td>Using the Microsoft Email Adapter</td>
</tr>
<tr>
<td>Microsoft SQL Server Adapter</td>
<td>Using the Microsoft SQL Server Adapter</td>
</tr>
<tr>
<td>MySQL Adapter</td>
<td>Using the MySQL Adapter</td>
</tr>
<tr>
<td>Oracle Commerce Cloud Adapter</td>
<td>Using the Oracle Commerce Cloud Adapter</td>
</tr>
<tr>
<td>Oracle CPQ Cloud Adapter</td>
<td>Using the Oracle CPQ Adapter</td>
</tr>
<tr>
<td>Oracle Database Adapter</td>
<td>Using the Oracle Database Adapter</td>
</tr>
<tr>
<td>Oracle Eloqua Cloud Adapter</td>
<td>Using the Oracle Eloqua Cloud Adapter</td>
</tr>
<tr>
<td>Oracle E-Business Suite Adapter</td>
<td>Using Oracle E-Business Suite Adapter</td>
</tr>
<tr>
<td>Oracle ERP Cloud Adapter</td>
<td>Using the Oracle ERP Cloud Adapter</td>
</tr>
<tr>
<td>Oracle Field Service Adapter</td>
<td>Using the Oracle Field Service Cloud Adapter</td>
</tr>
<tr>
<td>Oracle HCM Cloud Adapter</td>
<td>Using the Oracle HCM Cloud Adapter</td>
</tr>
<tr>
<td>Oracle JD Edwards EnterpriseOne Adapter</td>
<td>Using the JD Edwards EnterpriseOne Adapter</td>
</tr>
<tr>
<td>Oracle Logistics Adapter</td>
<td>Using the Oracle Logistics Cloud Adapter</td>
</tr>
<tr>
<td>Oracle Monetization Cloud Adapter</td>
<td>Using the Oracle Monetization Cloud Adapter</td>
</tr>
<tr>
<td>Oracle Messaging Cloud Service Adapter</td>
<td>Using the Oracle Messaging Cloud Service Adapter</td>
</tr>
<tr>
<td>Oracle NetSuite Adapter</td>
<td>Using the Oracle NetSuite Adapter</td>
</tr>
<tr>
<td>Oracle Responsys Adapter</td>
<td>Using the Oracle Responsys Adapter</td>
</tr>
<tr>
<td>Oracle Service Cloud (RightNow) Adapter</td>
<td>Using the Oracle Service Cloud (RightNow) Adapter</td>
</tr>
<tr>
<td>Oracle Engagement Cloud Adapter</td>
<td>Using the Oracle Engagement Cloud Adapter</td>
</tr>
</tbody>
</table>
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 navigation pane, click Integrations.
2. On the Integrations page, click Create.

   The Create Integration - Select a Style dialog is displayed.
3. Select the type of integration pattern applicable to your business needs. See Understand Integration Patterns.

The Create New Integration dialog is displayed.

4. Enter the following information:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>What do you want to call your integration?</strong></td>
<td>Provide a meaningful name so that others can understand the integration. You can include English alphabetic characters, numbers, underscores, and dashes in the identifier.</td>
</tr>
<tr>
<td><strong>Identifier</strong></td>
<td>Accept the default identifier value. The identifier is the same as the integration name you provided, but in upper case.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Version</td>
<td>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 now 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.</td>
</tr>
<tr>
<td>What does this integration do?</td>
<td>Provide a meaningful description so that others can understand the integration.</td>
</tr>
<tr>
<td>Which package does this integration belong to?</td>
<td>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.</td>
</tr>
</tbody>
</table>

5. Click Create.

6. See the following sections based on your selection.

<table>
<thead>
<tr>
<th>If You Selected...</th>
<th>See...</th>
</tr>
</thead>
<tbody>
<tr>
<td>App Driven Orchestration</td>
<td>Create Application-Driven Orchestrated Integrations</td>
</tr>
<tr>
<td>Scheduled Orchestration</td>
<td>Create Scheduled Integrations</td>
</tr>
<tr>
<td>File Transfer</td>
<td>Create Scheduled Integrations</td>
</tr>
<tr>
<td>Basic Routing</td>
<td>Create Basic Routing Integrations</td>
</tr>
<tr>
<td>Publish to OIC</td>
<td>Create an Integration to Publish Messages to Oracle Integration</td>
</tr>
</tbody>
</table>
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.

- Create an Orchestrated Integration
- Define Inbound Triggers and Outbound Invokes
- Add Actions, Connections, and Artifacts Through an Inline Menu
- Define Ad-Hoc Mappings
- Import a Map File into an Orchestrated Integration
- Add Actions to an Orchestrated Integration
- Use XPath Axis and Wildcard Expressions in the Expression Builder
- Use Lookups in Variable Assignments
- Define Fault Aggregation in Parallel Processing Scenarios
- Assign Business Identifiers for Tracking Fields
- Display Errors and Warnings in an Integration

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 Engagement Cloud...
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

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.
  – **Zoom Out**: Click to decrease 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.
  – **Home**: Click to align the integration in normal size in the upper left corner of the page.
  – **Start**: Click to reset the integration to its normal size and place it in the middle of the page.
  – **Multiple Select**: Click and then drag the cursor around parts of the integration to select them. This action highlights the selected sections in blue.
  – **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 your 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.

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.
<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Commonly Used</td>
<td>Displays actions that are commonly placed between the two nodes.</td>
</tr>
<tr>
<td>Connections</td>
<td>Displays adapter connections available to add. The option is only displayed when a connection can be added after the nodes.</td>
</tr>
<tr>
<td>Actions</td>
<td>Displays actions that are allowed to be placed between the two nodes.</td>
</tr>
<tr>
<td>Integration Artifacts</td>
<td>Displays integration artifacts that are allowed to be placed between the two nodes.</td>
</tr>
</tbody>
</table>

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.
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 to create two mappings for the defined and otherwise branches that output to the order invoke connection.

2. On the right side, expand the **Actions** section.

3. Drag a **Map** icon to a branch of the switch activity. For this example, the map is first dragged to the + sign of the defined branch.

The Select Output dialog is displayed.
4. Select the endpoint to which to map. For this example, the **Order** endpoint is selected.

The mapper is displayed.

5. Map appropriate elements from the source data structure to the target data structure.

6. When complete, click **Close**, then click **Apply** to save your changes.

7. Drag a **Map** icon to the otherwise branch of the switch activity.

The Select Output dialog is displayed.

8. Select the endpoint to which to map, and click **Select**. For this example, the **Order** endpoint is also selected.

The mapper is displayed.

9. Map appropriate elements from the source data structure to the target data structure.

10. When complete, click **Close**, then click **Apply** to save your changes.

Mappings for both branches of the switch activity are now displayed.

- The inbound trigger connection is configured with an opportunity business object.
- A switch activity is defined with two branches:
  - If the routing expression for the defined branch resolves to true, the mapping for that data is performed for the outbound invoke connection.
  - If the routing expression for the defined branch does not resolve to true, the otherwise branch is taken and the mapping for that data is performed for the outbound invoke connection.

You can click the mappers to display a menu for editing or viewing the mapper. You can also click the invoke connections to edit or view their contents.
You can make the integration as complex as is required. For example, you can:

- Add additional switches to other parts of the integration.
- Add additional ad-hoc mappings to the branches of the switches.

**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. Once imported, the map file cannot be edited. 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
- Route Expressions with Switch Branches
- Manage a Group of Actions and Fault Handlers with a Scope Action
- Assign Values to Scalar Variables in an Assign Action
- Loop Over Actions or Invoke Connections While a Condition is Satisfied with a While Action
- Send Notification Emails During Stages of the Integration with a Notification Action
- Delay Integration Execution for a Specified Time Period with a Wait Action
- Add Global Fault Handling to Orchestrated Integrations
- Catch Faults with a Re-throw Fault Action
- Throw Faults with a Throw New Fault Action
- Process Files in Scheduled Integrations with a Stage File Action
- Log Messages with a Logger Action
- Add a JavaScript Action
- Add Placeholder Notes with a Note Action

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. On the right side of the canvas, click Actions to expand the panel.
2. Drag the For Each icon to the plus sign where you want to loop over an element. The For Each dialog is displayed.
3. Expand the Source tree to select an element.
4. 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**.

5. 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.

6. Enter an alias for the current file of the iteration in the **Current Element Name** field.

7. Click **Done**.

The for-each action is displayed in the canvas. A looping arrow indicates that this action performs repetitive looping.
8. Drag other actions inside the for-each action to define what should happen during each iteration of the loop. For this example, an assign action is dragged to the plus sign below the for-each action.

Enter a name for the assign action, and click OK.

9. Click the Edit icon.

The Expression Builder is displayed. Note that the value you entered in the Current Element Name field in Step 6 (currentFileName) is displayed as an expandable element in the Source tree.

10. Expand $currentFileName in the Source tree.

The repeatable element you added in Step 5 (ICSFile) is displayed.

11. Drag the ICSFile element to the Expression field, and click Close.

12. In the Assign dialog, enter a variable name in the Name field, and click Exit Assignments.

The assign action is added to the scope of the for-each action. After integration activation, the for-each action loops over the assign action for each downloaded file.

Tracking the Status of a For-Each Action During Runtime

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

1. In the navigation pane, select Monitoring > 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. In the upper right corner, click the Actions menu and select View Audit Trail.
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.

Route Expressions with Switch Branches

You can define switch branches to add routing expressions in your integration.

To define switch branches:

1. On the right side of the canvas, click **Actions** to expand the panel.
2. Drag the **Switch** icon to the integration canvas. As you do, large + sections within circles are displayed that indicate where you can drop the switch activity. For this example, the switch activity is added immediately after the trigger connection.

   **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.

3. Click the **Undefined** branch icon.
4. Select the **Edit** icon from the menu that is displayed. This invokes the Expression Builder.
5. 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 non-boolean values are not supported.

   For this example, the following expression is defined:

   
   \[ /\text{nsrcmpr}:\text{process}/\text{nsrcmpr}:\text{Contact\_CustomObj}/\text{rnb\_v1\_3}:\text{LookupName} = "Denver" \]

   You can now define different data flows for both the defined and otherwise branches in the switch activity.

6. Return to the left side of the canvas, and click **Invokes** to expand the panel.
7. Drag an adapter to the appropriate + section within the circle. Two are available: one for the defined branch and another for the otherwise branch. For this example, the section for the otherwise branch is selected.

![Diagram](image)

This invokes the Adapter Endpoint Configuration Wizard.

8. Complete the pages of the wizard to configure the selected adapter.

This creates an extra invoke connection on the otherwise branch. You can use the data from this invoke connection to initiate an order with the endpoint.

![Diagram](image)

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

Creating a Scope Action

To create a scope action:

1. On the right side of the canvas, click **Actions** to expand the panel.
2. Drag the **Scope** icon to the appropriate **plus** sign location in the integration.
3. Enter a name and optional description when prompted, and click **Create**.
   The action is displayed in the integration.

4. 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.
5. 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.

6. 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.
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 audit trail for an activated integration. This is only possible if there is a tracking instance.

1. In the navigation pane, select Monitoring > 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. In the upper right corner, click the menu and select **View Audit Trail**. Details about processing status are displayed. For example:

   - If scope execution succeeds, details similar to the following are displayed:

     ```
     Message processed successfully for Trigger GetWeather.
     Message entered Scope Scope.
     Message processed successfully for Invoke GetOrg.
     Message exited Scope Scope.
     Message processed successfully for Reply GetWeather.
     ```

   - If scope execution fails, but the fault handler succeeds in handling the fault, details similar to the following are displayed:

     ```
     Message processed successfully for Trigger GetWeather.
     Message entered Scope Scope.
     Error processing message in Invoke GetOrg.
     Scope fault handler started.
     Scope fault handler ended.
     Message exited Scope Scope.
     Message processed successfully for Reply GetWeather.
     ```
• If both the scope and fault handler fail, details similar to the following are displayed:

Message processed successfully for Trigger GetWeather.
Message entered Scope Scope.
Error processing message in Invoke GetOrg.
Scope fault handler started.
Error processing message in Invoke faultInvoke
OSB-380001: Invoke JCA outbound service failed with application error, except...

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

Assign Values to Scalar Variables in an Assign Action

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

1. On the right side of the canvas, click Actions to expand the panel.
2. Drag the Assign icon to the appropriate plus sign location in the integration canvas.
3. Enter an assignment name and optional description when prompted by the Assignments dialog, then click OK.
4. 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:

   • 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.

5. 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.
6. Click **Yes** when prompted to save your changes. The expression value is displayed in the **Value** field.

7. Click the **Add** icon to add multiple assignments to the **Assign** node. You can also define an assignment with a value from a previously defined assignment. In the following example, **AccountId** (the previously created assignment) is assigned as the value of assignment **AccountIdNow**.

8. Click **Exit Assignments**.
9. Click **Save**.

Variable assignments can be of greater complexity. For example, you can use assignments in switch activities and in maps. For example, in the following orchestration, the next organization ID is assigned.
1. Click the ASSIGN node. **NewOrgID** is assigned a value of **OrganizationId + 1.0**. **OrganizationId** is the business object that comes from the trigger element that initiates this orchestration.

2. View the switch activity in the orchestration. While **OrganizationId** is less than or equal to 15.0, another assignment is performed in **VarNodeInSwitch**.
3. Click **VarNodeInSwitch** and note that it reverts to the original value.

You can also map assignment values in the mapper.

If there is an error in your assignments, an identifier is displayed next to the **ASSIGN** node and also in the mapper.

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.

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 must be of a number type, and not
a string type. Otherwise, they must be explicitly cast to number.

1. On the right side of the canvas, click Actions to expand the palette.
2. Drag the While icon to the Plus sign where you want to create a looping condition.
3. Enter a name and optional description for the while action when prompted, then
click OK.
   The Expression Builder is displayed.
4. Expand the Source tree to select an element. You can also add functions and
   operations.
5. Drag an element 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.

6. Click Exit Condition Builder, then click Yes when prompted to save your
changes. The values of the while action are displayed in the canvas.
7. 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.

8. 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.
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 audit trail for an activated integration. This is only possible if there is a tracking instance.

1. In the navigation pane, click Integrations, then click the arrow next to Designer.
2. Click Monitoring, then click Tracking.
3. 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.
4. In the upper right corner, click the menu and select View Audit Trail.
   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.

Message processed successfully for Trigger trigger
Message processed successfully for Assignment init
Message processed successfully for Invoke invoke
Message processing started for While outerwhile
Message processing ended for While outerwhile
Message processed successfully for Reply trigger

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 also create the body part of an email using parameters defined in the Expression Builder.

1. On the right side of the canvas, click Actions to expand the panel.
2. Drag the Notification icon to the Plus sign where you want to add a notification. For example, you may define notifications on the two branches of a switch action.
3. Enter a name and optional description for the notification action when prompted, then click OK.
4. 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**.

5. 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.

6. To add a parameter name and value, click the **Plus** icon in the **Parameters** section.

7. Enter a parameter name and description.

8. Click the **Expression Builder** icons to define parameter values (for this example, **ID** and **LookupName** are created).

When complete, the page looks similar to the following:

![Notification page with parameters](image)

9. Click **Done**.

When the email notification is received during integration runtime, the two parameters of **ID** and **LookupName** are replaced with dynamic values:

All recipients;

This is an automatically generated e-mail from the **Complex_Switch_REF_Trigger** orchestration.

The ID passed is 20. The corresponding LookupName is **TestOrg1**.

BR,
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 LookupName or ID@id examples used in this section are modified, the parameter assignment become invalid. Or when $lastname or @firstname is deleted, the To field becomes 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 Tracking 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:

```
* From  $from
* To    concat( $to, ",", $SenderRecipient)  
* Subject concat( $subject, ">", DataFileName, ", File Failed To Load To Fusion")
```

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. On the right side of the canvas, click Actions to expand the palette.
2. Drag the Wait icon to the Plus sign where you want to delay execution of the integration.
3. 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.
4. 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 audit trail for an activated integration. This is only possible if there is a tracking instance.

1. In the navigation pane, select Monitoring > Tracking.
2. Click the business identifier value of the integration to track.
   The integration flow (including any wait actions) is displayed.
3. In the upper right corner, click the menu and select View Audit Trail.
   Details about processing status (including any wait actions) are displayed. For example:

   Schedule triggered successfully.
   Wait one_min triggered.
   Message processed successfully for Invoke weather.
   Processing completed successfully.

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. From the menu in the upper right corner, select **Global Fault**.

The Global Fault Handler page is displayed. The initial trigger in your integration is automatically connected to an initial **Error Hospital** action that cannot be deleted. However, you can add and delete other **Error Hospital** actions. The **Error Hospital** action does not respond back to the trigger. Instead, details collected by the **Error Hospital** 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 **Error Hospital** actions, as needed.

3. Select the **Actions** palette, and drag a **Switch** action into the integration.

4. Click the **Undefined** 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 > ICS for designing fault handling:

- getFaultAsString (returns the fault as a string value)
- getFaultAsXML (the fault as an XML element)
- getFaultName (returns the fault name)

These functions are only available within the Expression Builder in orchestrated integrations.

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

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

6. Click Close to return to the Global Fault Handler page.

7. From the Actions palette, drag a Stop action to the Otherwise branch of the switch action.

The fault handling logic is now complete. For the above example, if an error occurs, error details are captured and sent to the error handling framework (as indicated by the Error Hospital action). If no error occurs, the fault handling stops and nothing is sent to the error handling framework (as indicated by the Stop action).

The Delete icon in the upper right corner of the page enables you to delete the designed faulting handling logic. You can also set integration tracking business identifiers on either the Global Fault Handling page or the integration canvas page.
8. Click **Save**.

9. From the menu in the upper right corner, select **Orchestration** to return to the integration canvas.

10. Complete design of the orchestrated integration.

11. Click **Save**, then click **Close**.

12. 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 audit trail for an activated integration. This is only possible if there is a tracking instance.

1. In the navigation palette, select **Monitoring > 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. From the menu in the upper right corner, select **View Global Fault**.

   The fault handling logic you designed is displayed in view-only mode.

4. Click **Done**.

5. From the menu in the upper right corner, select **View Audit Trail**.

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

   ```
   Message processed successfully for Trigger trigger.
   Message routed through Switch Route Otherwise.
   Error processing message in Invoke invoke2
   Global fault handler triggered.
   In Global Fault Handler Message routed through Switch Route Otherwise
   In Global Fault Handler Message processed successfully for Reply trigger
   
   If the global fault handler successfully handled the error, the integration is displayed as **COMPLETED** on the Tracking 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. Create an integration that includes a scope action. See Manage a Group of Actions and Fault Handlers with a Scope Action.

2. Click Fault Handler on the right side of the scope.

3. On the right side of the canvas, click Actions to expand the panel.

4. 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.

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

The Re-throw Fault icon is hidden.

6. Design additional logic in the scope action.

7. 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. On the right side of the canvas, click Actions to expand the palette or place your cursor at the appropriate place in the integration and click the + sign to invoke the inline menu.

2. Drag the Throw New Fault icon into the integration or select it from the inline menu.

   The Create Action dialog is displayed.

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

   The Throw New Fault page is displayed.

4. Define the action:

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code</td>
<td>Click the Edit icon to create an expression in the Expression Builder. This is a mandatory field.</td>
</tr>
<tr>
<td>Reason</td>
<td>Click the Edit icon to define a reason for the error in the Expression Builder.</td>
</tr>
<tr>
<td>Details</td>
<td>Click the Edit icon to define additional error details in the Expression Builder.</td>
</tr>
<tr>
<td>Skip Condition</td>
<td>Define a condition to prevent the fault from being throw in the Skip Condition version of the Expression Builder.</td>
</tr>
</tbody>
</table>

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 references to files 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 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 Integration folders 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.

1. On the right side of the canvas, click Actions to expand the panel.
2. Drag the Stage File icon to the plus sign where you want to perform file actions.
   The Configure Stage File Action wizard is displayed.
3. 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.

1. From the dropdown list, select the stage file operation to perform. The page refreshes to display fields appropriate to your selection.

   - List File
   - Read Entire File
   - **Read File in Segments** (for chunking files)

   ![Note:]

   If you select this operation, note the following restrictions when specifying a schema later in the Configure Stage File Action wizard:

   - 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

   - You cannot upload a ZIP file that includes multiple schemas in which one schema has a reference to the other schema file (for example, schema A imports schema B). In this case, the **Read File in Segments** option does not work when it reads the element that belongs to schema B.

   - Unzip File
   - Write File
   - Zip File
### Table 3-1  - List File

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify the Directory to List Files from</td>
<td>Click the <strong>Expression Builder</strong> icon to build an expression to specify the directory from which to list files.</td>
</tr>
<tr>
<td>Specify the File Pattern to use</td>
<td>Specify the pattern of the file name to transfer to the output directory. Click the ? icon for the list of supported patterns.</td>
</tr>
<tr>
<td>List Files Recursively</td>
<td>Select to list the files recursively.</td>
</tr>
</tbody>
</table>

### Table 3-2  - Read Entire File

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure File Reference</td>
<td>• <strong>Yes</strong>: Select to process an upstream operation that provides a file reference. Once selected, you specify the file reference.</td>
</tr>
<tr>
<td></td>
<td>• <strong>No</strong>: Select to process a file name.</td>
</tr>
<tr>
<td>Specify the File Reference</td>
<td>Click the <strong>Expression Builder</strong> icon to specify a file reference.</td>
</tr>
<tr>
<td>Specify the File Name</td>
<td>Click the <strong>Expression Builder</strong> icon to build an expression to specify the file name (for example, /nssrcmpr:schedule/nssrcmpr:startTime).</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: The file size must be less than 10 MB. For files greater than 10 MB, use the <strong>Read File in Segments</strong> operation.</td>
</tr>
<tr>
<td>Specify the Directory to read from</td>
<td>Click the <strong>Expression Builder</strong> icon to build an expression to specify the directory from which to read files.</td>
</tr>
<tr>
<td>Remove trailer</td>
<td>Select to not remove the trailer, to remove the last row, or to remove the last n rows.</td>
</tr>
</tbody>
</table>

### Table 3-3  - Read File in Segments

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure File Reference</td>
<td>• <strong>Yes</strong>: Select to process an upstream operation that provides a file reference.</td>
</tr>
<tr>
<td></td>
<td>• <strong>No</strong>: Select to process a file name.</td>
</tr>
<tr>
<td>Specify the File Reference</td>
<td>Click the <strong>Expression Builder</strong> icon to specify a file reference.</td>
</tr>
</tbody>
</table>
Table 3-3  (Cont.) - Read File in Segments

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify the File Name (Appears if you select No for the Configure File Reference.)</td>
<td>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.</td>
</tr>
<tr>
<td>Specify the Directory to read from</td>
<td>Click the Expression Builder icon to build an expression to specify the directory from which to read.</td>
</tr>
</tbody>
</table>
| 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 segments sequentially.                                                                                                                                                           |

Table 3-4  - Unzip File

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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. Click the Expression Builder icon to specify a ZIP file reference.                                                                                                    |
| Specify the File Reference (Appears if you select Yes 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 Name (Appears if you select No for the Configure File Reference.) | Click the Expression Builder icon to build an expression to specify the directory in which to unzip the file.                                                                                              |
| Specify the Zip File Directory        |                                                                                                                                             |
Table 3-4  (Cont.) - Unzip File

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify the Directory to Unzip</td>
<td>Click the <strong>Expression Builder</strong> icon to build an expression to specify the directory in which to unzip files.</td>
</tr>
</tbody>
</table>

Table 3-5  - Write File

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify the File Name</td>
<td>Click the <strong>Expression Builder</strong> icon to build an expression to specify the file name.</td>
</tr>
<tr>
<td>Specify the Output Directory</td>
<td>Click the <strong>Expression Builder</strong> icon to build an expression to specify the output directory.</td>
</tr>
<tr>
<td>Append to Existing File</td>
<td>Optionally select to append records to the existing file.</td>
</tr>
</tbody>
</table>

Table 3-6  - Zip Files

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify the File Name</td>
<td>Click the <strong>Expression Builder</strong> icon to build an expression to specify the file name.</td>
</tr>
<tr>
<td>Specify the Directory to zip</td>
<td>Click the <strong>Expression Builder</strong> icon to build an expression to specify the directory to ZIP.</td>
</tr>
<tr>
<td>Specify the Output Directory</td>
<td>Click the <strong>Expression Builder</strong> icon to build an expression to specify the output directory in which to write a ZIP file.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do you want to specify the structure for the contents of the file for the payload?</td>
<td>Select <strong>Yes</strong> to define a schema format to use for the files to transfer. Select <strong>No</strong> if a schema is not required and you want to send opaque files (for example, a GIF or PNG file).</td>
</tr>
</tbody>
</table>
Which one the following choices would be used to describe the structure of the file contents?

Select an option:

- **Sample delimited document (e.g. 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 NameSpace)**: Select to create a new schema from a sample XML file with a single or no namespace.
- **Sample JSON document**: Select to create a new schema from a JSON file.

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 3-7  - Sample Delimited Document (e.g. CSV)

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select the Delimited Data File</td>
<td>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.</td>
</tr>
<tr>
<td>Enter the Record Name</td>
<td>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.</td>
</tr>
<tr>
<td>Enter the Recordset Name</td>
<td>Enter the recordset name. This becomes the root element of the created schema file.</td>
</tr>
</tbody>
</table>
### Table 3-7 - Sample Delimited Document (e.g. CSV)

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select the Field Delimiter</td>
<td>Select one of the following supported file delimiter options:</td>
</tr>
<tr>
<td>• Single space</td>
<td>• Comma</td>
</tr>
<tr>
<td>• Semicolon</td>
<td>• Tab</td>
</tr>
<tr>
<td>• Pipe (for example, Name</td>
<td>City</td>
</tr>
<tr>
<td>Character Set</td>
<td>Select a character set. The selected value will be used as the encoding format while reading the sample data file.</td>
</tr>
<tr>
<td>Optionally Enclosed By</td>
<td>This value causes occurrences of the selected delimiter to be ignored during processing. For example, when processing the following record:</td>
</tr>
<tr>
<td></td>
<td>Fred,&quot;2 Old Street, Old Town,Manchester&quot;, 20-08-1954,0161-499-1718</td>
</tr>
<tr>
<td></td>
<td>If the selected <strong>Field Delimiter</strong> is &quot;,,&quot; and the <strong>Optionally Enclosed By</strong> value is quot;(&quot;), then the value 2 Old Street, Old Town,Manchester is treated as a single record column.</td>
</tr>
<tr>
<td>Use the First Row as Column Headers</td>
<td>Displays by default the first row of the selected CSV file as the column headers.</td>
</tr>
<tr>
<td>Detach</td>
<td>Select to edit the CSV file in a separate window.</td>
</tr>
<tr>
<td>Mark All As Optional</td>
<td>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.</td>
</tr>
</tbody>
</table>

### Table 3-8 - XML Schema (XSD) Document

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select a New File</td>
<td>Select an existing XML schema file or schema archive file from the file system.</td>
</tr>
<tr>
<td>Selected File Name</td>
<td>Displays the selected schema file name.</td>
</tr>
</tbody>
</table>
Table 3-8  (Cont.) - XML Schema (XSD) Document

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Select the Schema Element</td>
<td>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.</td>
</tr>
<tr>
<td>• Select Repeating Batch Element</td>
<td>Click the Expression Builder icon to build an expression to identify the repeating node in the schema to support payload chunking.</td>
</tr>
</tbody>
</table>

Table 3-9  - Sample XML Document (Single or No Namespace)

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Select a New File</td>
<td>Select a sample XML document from the file system. The file must contain no namespace or a single namespace.</td>
</tr>
<tr>
<td>• Selected File Name</td>
<td>Displays the selected schema file name.</td>
</tr>
<tr>
<td>• Select the Schema Element</td>
<td>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.</td>
</tr>
<tr>
<td>• Select Repeating Batch Element</td>
<td>Click the Expression Builder icon to build an expression to identify the repeating node in the schema to support payload chunking.</td>
</tr>
</tbody>
</table>

Table 3-10  - Sample JSON Document

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Select a New File</td>
<td>Select a sample JSON document from the file system.</td>
</tr>
<tr>
<td>• Selected File Name</td>
<td>Displays the selected schema file name.</td>
</tr>
<tr>
<td>• Select the Schema Element</td>
<td>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.</td>
</tr>
</tbody>
</table>

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).
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. On the right side of the canvas, click **Actions** to expand the palette.
2. Drag the **Logger** icon to the **plus** sign where you want to begin logging.
3. Enter a name and optional description, then click **OK**.

   The Logger page is displayed.

4. 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**.

5. 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)
   ```

6. 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)
   ```

7. When complete, click **Close**. For this example, two loggers are 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 navigation pane, click **Integrations**, then click the < arrow next to **Designer**.
2. Click **Monitoring**.
3. Click **Activity Stream** to view the messages.

```
FILE_TRANSFER_SAMPLE - LogActivity - Thu May 25 16:03:11 PDT 2017 - The
file 1KB170525160309105.zip has been uploaded to /upload 400028
FILE_TRANSFER_SAMPLE - LogActivity - Thu May 25 16:03:07 PDT 2017 -
Filename is: 1KB.zip 400028
```
4. Click **Download Diagnostic Logs** to view the messages.
5. Open the zip file and browser to **ics_server1/ics-flow.log** to 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:

1. On the right side of the canvas, click **Actions** to expand the palette.
2. Drag the **Javascript** icon to the **Plus** sign where you want to add a function.
3. Enter a name and optional description for the JavaScript action when prompted, then click **OK**.

    You can use the `/` menu to display information about the Javascript or to delete the JavaScript and return to the **Actions** palette.

4. Click the **+Function** button.

    The Select a Function popup appears.

5. Click a function and click the **Select** button in the function’s row.

    The configuration page is displayed. It shows the details of the selected function including the input and output parameters.

6. Click the pencil icon in the **Value** column to use the Expression Builder to configure the input parameters.
7. Click **Validate** in the title bar to validate the parameters.
8. 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 audit trail for an activated integration. This is only possible if there is a tracking instance.

1. In the navigation pane, click **Integrations**, then click the < arrow next to **Designer**.
2. Click **Monitoring**, then click **Tracking**.
3. Click the business identifier value of the integration to track.
4. The integration (including any JavaScript actions) is displayed. Any JavaScript action failures are identified by red arrows.

5. In the upper right corner, click the menu and select **View Audit Trail**.

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

- Message processed successfully for Trigger FF_SOAP
- Message processed successfully for Callout callout1
- Message processed successfully for Assignment callout_return
- Processing completed successfully.

**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. On the right side of the canvas, click **Actions** to expand the palette.
2. Drag the **Note** icon to the location where you want to add a note. The Create Action dialog is displayed.
3. Enter a name and description of the action, then click **OK**. The action is added to the integration.
4. Click the **Edit** icon to add your notes. You can add up to 256 characters.
5. Enter your notes, then click **OK**.
6. Hover your cursor over the icon to display the note text.
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:

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>child::book</code></td>
<td>Selects all <code>book</code> nodes that are children of the current node.</td>
</tr>
<tr>
<td><code>attribute::lang</code></td>
<td>Selects the <code>lang</code> attribute of the current node.</td>
</tr>
<tr>
<td><code>child::*</code></td>
<td>Selects all element children of the current node.</td>
</tr>
<tr>
<td><code>attribute::*</code></td>
<td>Selects all attributes of the current node.</td>
</tr>
<tr>
<td>Syntax</td>
<td>Result</td>
</tr>
<tr>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td><code>child::text()</code></td>
<td>Selects all <code>text</code> node children of the current node.</td>
</tr>
<tr>
<td><code>child::node()</code></td>
<td>Selects all children of the current node.</td>
</tr>
<tr>
<td><code>descendant::book</code></td>
<td>Selects all <code>book</code> descendants of the current node</td>
</tr>
<tr>
<td><code>ancestor::book</code></td>
<td>Selects all <code>book</code> ancestors of the current node.</td>
</tr>
<tr>
<td><code>ancestor-or-self::book</code></td>
<td>Selects all <code>book</code> ancestors of the current node and the current if it is a <code>book</code> node.</td>
</tr>
<tr>
<td><code>child::*/child::price</code></td>
<td>Selects all <code>price</code> grandchildren of the current node.</td>
</tr>
</tbody>
</table>

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).

<table>
<thead>
<tr>
<th>SOAP (Adapter)</th>
<th>SOAPCITY (Domain Name)</th>
</tr>
</thead>
<tbody>
<tr>
<td>80112</td>
<td>Englewood</td>
</tr>
<tr>
<td>85003</td>
<td>Phoenix</td>
</tr>
<tr>
<td>80007</td>
<td>Arvada</td>
</tr>
<tr>
<td>80220</td>
<td>Denver</td>
</tr>
</tbody>
</table>

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 > ICS**.
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.

![Lookup Table](image)

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.

![Variable Assignment](image)

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, then click **Save**.

![Integration Mapper](image)

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:

<table>
<thead>
<tr>
<th>Location</th>
<th>Append Option Allowed?</th>
<th>Clear Option Allowed?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top level (that is, outside of any</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>looping actions)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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.
Assign Business Identifiers for Tracking Fields

Business identifiers enable you to track payload fields in messages during runtime. You must assign business identifiers before you can activate an orchestrated integration.

1. From the menu in the upper right corner, select Tracking to assign business identifiers for tracking fields. See Assign Business Identifiers for Tracking Fields in Messages.
2. When complete, click Done.
3. When the orchestrated integration is complete, click Save, then click Close.
4. Activate your integration. See Activate an Integration.
5. Manage and monitor your integration. See Monitor Integrations and Track Business Identifiers in Integrations During Runtime.

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.

Define Inbound Triggers and Outbound Invokes

To define inbound triggers and outbound invokes:

1. On the right side of the canvas, click Triggers to expand the panel.

   If the adapter connection you want to use does not exist, click the + sign 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. Click the adapter type to display the specific type and number of configured adapters. Synchronous, asynchronous, and fire-and-forget (no response) triggers are supported.

3. Drag the configured adapter to the large + section within the circle in the integration canvas.

   This invokes the Adapter Endpoint Configuration Wizard.

4. Complete the pages of the wizard to configure the selected adapter. For this example, an Oracle Service Cloud (RightNow) Adapter is selected in which a request opportunity business object and an immediate 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).
Asynchronous responses are also supported. You can select the **Delayed** (asynchronous) option on the Response page in the Adapter Endpoint Configuration Wizard for the Oracle Service Cloud (RightNow) Adapter. Creating an asynchronous response creates a return (**Receive**) activity in the integration. On the right side, the **Triggers** section is replaced by an **Invokes** section that enables you to add multiple outbound invoke connections to the integration.

An **Actions** section is now displayed below **Invokes**. When expanded, this section displays the following options:

- **Assign**: Enables you to assign variables to integrations.
- **Javascript**: Enables you to add JavaScript functions to the integration.
- **Logger**: Enables you to add log messages to the activity stream and diagnostic logs.
- **Map**: Enables you to add ad-hoc mappers to the integration.
- **Notification**: Enables you to send a notification email to relevant users at specific points in the execution of an integration.
- **Scope**: Enables you to manage a collection of child actions and invokes that can have their own fault handlers.
- **Stage File**: Enables you to process files in scheduled integrations.
- **Switch**: Enables you to add a switch activity for defining routing expression branches in the integration.
• **For Each**: Enables you to loop over a repeating element and execute one or more actions within the scope of the for-each action.

• **While**: Enables you to loop over actions or invoke connections as long as a specific condition is met.

• **Raise Error**: Enables you to send failed messages to the error hospital for further analysis.

• **Fault Return**: Enables you to return a fault.

• **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.

• **Return**: Enables you to return an immediate response.

• **Stop**: Enables you to terminate the integration. No response message is returned to the trigger.

• **Wait**: Enables you to delay the execution of an integration for a specified period of time.

5. On the right side of the canvas, click **Invokes** 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.

6. Click the adapter type to display the specific type and number of configured adapters.

7. 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 (this is similar to the enrichment feature that you can define in integrations that are not orchestrated).
   - A section after the request mapper.

8. 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.

9. Complete the pages of the wizard to configure the selected adapter. For this example, an Oracle Engagement Cloud 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.

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:

• Create a Basic Routing Integration
• Add a Trigger (Source) Connection
• Add an Invoke (Target) Connection
• Add Request and Response Enrichments
• Delete Request and Response Enrichments
• Create Routing Paths for Two Different Invoke Endpoints in Integrations
• Create Routing Expression Logic in Both Expression Mode and Condition Mode
• Delete Routing Paths

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 Designer, drag a connection from the Connections or Technologies panel on the right to the Trigger (source) area on the canvas.
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 Designer, drag a connection from the Connections or Technologies panel on the right to the **Invoke** (target) area on the canvas.

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:

<table>
<thead>
<tr>
<th>Mapping</th>
<th>Source</th>
<th>Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>Request</td>
<td>HelloRequest/FirstName</td>
<td>sayHello/name</td>
</tr>
<tr>
<td>Response</td>
<td>sayHelloResponse/sayHelloReturn</td>
<td>HelloResponse/Greeting</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Mapping</th>
<th>Source</th>
<th>Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>Response Enrichment</td>
<td>sayHelloResponse/sayHelloReturn</td>
<td>visitTimestampReq &gt; reqMsg</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Mapping</th>
<th>Source</th>
<th>Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>Response</td>
<td>$ResponseEnrichmentApplicationObject &gt; visitTimestampResp &gt; respMsg</td>
<td>HelloResponse/Greeting</td>
</tr>
</tbody>
</table>

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 Integration 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.

```
Routing Expression: /nssrcmpr:process/nssrcmpr:Organization/tns:ClassificationCode = 'Org'
```

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.

17. Click the Request Mapper icon to define the mapping.
   For this example, the following mapping is defined.

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>process &gt; Organization &gt; Organizationid</td>
<td>Get &gt; Account &gt; ID &gt; id</td>
</tr>
</tbody>
</table>

18. Click the Response Mapper icon to define the mapping.
   For this example, the following mapping is defined.

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>process &gt; GetResponse &gt; Account &gt; ID &gt; LookupName</td>
<td>processResponse &gt; Organization &gt; Name</td>
</tr>
</tbody>
</table>

Integration design is 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.

![Expression Name](image)

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.

![Select an element and move.](image)

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.

![Expression Name](image)

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:**

   - 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.

![Expression Name](#)

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.

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 map data
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

- Create a Scheduled Integration
- Create a Schedule
- Create Parameters in Scheduled Integrations
- Start and Pause an Integration Schedule
- View Past and Present Integration Runs
- View Future Runs
- Edit an Integration Schedule
- Delete an Integration Schedule
- Unlock an Integration Schedule
- Delete Runs
- Monitor Integration Runs
- Create Ad-Hoc Integration Runs Using the REST API
- Activate Older, Imported Scheduled Integrations Using Operations Other Than Read Multiple Files

See Getting Started with the FTP Adapter.

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. If you click the Schedule icon, you can select the Edit Schedule Definition icon that invokes the Schedule Definition page for defining the schedule details.

2. See Create an Orchestrated Integration to design your scheduled integration. The following sections describe how to configure schedule runs.

Create a Schedule

You can create a schedule for running orchestrated integrations.

To create an integration schedule:

Note:

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 Integration page.

2. Find the integration in which trigger and invoke FTP adapter connections have been defined.
   These integrations are identified by a Schedule icon that depicts a calendar. There are several ways in which to create an integration.

3. If you want to first activate the integration and then create the schedule:
   a. Click the Activate icon.
   b. Click Activate and Schedule.

4. If you want to first create a schedule for an integration that you activate later:
   • Select Add Schedule from the dropdown menu at the far right.

5. In the Schedule Name field, accept the default name or click the Edit icon to enter a new name. When complete, enter an optional description and click the green checkmark. This action does not save the changes.

   **Note:**
   Assume you create a schedule and generate several integration runs, then deactiviate 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 Runs page, the past runs of the deleted integration are displayed. This is by design.

6. If you want to schedule basic integration runs, click Basic. 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.

7. If you want to schedule integration runs with an iCal expression, click Advanced.
   • 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.

     ```
     FREQ=MONTHLY;BYDAY=1,10,15;BYHOUR=5,10,15,20,30;BYMINUTE=30,40,50;BYSECOND=0
     &FREQ=MONTHLY;BYDAY=1,10,15;BYHOUR=18;BYMINUTE=0,10,20,30,40,50;BYSECOND=0
     &FREQ=MONTHLY;BYDAY=1,10,15;BYHOUR=19;BYMINUTE=0,10,20,30;BYSECOND=0
     ```

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

iCal expression is valid.

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.

8. In the Frequency section, 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.

9. Click the green checkmark icon for each frequency type that you specify.

The Time zone field shows the value you selected on the Preferences page.

10. Click Save to validate your frequency settings.

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

11. 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.

12. If you want to start the integration run when the schedule is activated:
   
   • Click When schedule starts.

13. 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.

14. In the Until section, click the link to the right.

A menu is displayed for defining the expiration date.
15. If you want the schedule run to never expire:
   • Select **Never (repeat indefinitely)**.

16. 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**.

17. Click **Save**.
   If successful, a message is displayed in the upper left corner.

   *Schedule Run name saved successfully*

18. Click **Close**.
   The Schedule and Future Runs page is displayed and the Monitoring tab is highlighted in the upper right corner.
   Details about the integration run schedule are displayed. You can click the **Edit** icon to update the schedule definition.

19. If you have already activated the integration, select an option to run the integration:
   a. Select **Submit Now** from the menu.
   b. 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).
   c. Click **Submit Now**.
   or
   a. Select **Start Schedule** on the Schedule and Future Runs page to activate the integration schedule.
   b. 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.

20. If you have not yet started the integration, return to the Integrations page and click **Activate**.

21. Run the integration as described in the previous steps.

Create 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.

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** pattern.
3. Complete the fields of the dialog, and click **Create**.
   - The integration canvas is displayed.
4. Click the **Schedule** icon, then select **Edit**.
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 *position* and a value of 20 are entered.

7. Click the **plus** icon, and enter a name (for this example, *region* is entered) and optional description, but leave the **Value** column empty.

8. Click **Close** to save your changes and exit the page.
   
   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.

9. From the **Actions** section, drag an **Assign** icon into the integration, enter a name, and then click **OK**.

10. In the **Name** column, click the **plus** icon, then select the parameter from the dropdown list that you created without a value in the Schedule Parameters page.

11. Click the **Edit** icon to invoke the Expression Builder.

12. Create an expression, then click **Close**.

13. In the Assign page, click **Close**.
   
   You are returned to the integration canvas.
14. 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-loop.

---

**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 Configure 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 for the specific integration on the Integrations page.
      - Select **Submit Now** from the menu on the Schedule and Future Runs page of the scheduled integration.
      - Click the **Submit Now** icon on the Monitor Integrations page.
   b. If scheduling an integration, select **Start Schedule** on the Schedule and Future Runs page.

The Configure Schedule Parameters dialog is displayed.
2. 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.

To start and pause an integration schedule:

1. Go to the Integration page.
2. Find the integration on which the scheduled run is defined.
3. Select **Schedule** from the menu at the right.
   The Schedule and Future Runs page is displayed.
4. Click the **Start Schedule** box.
5. Click **Start** when prompted to start the schedule.
   The following message is displayed at the top of the page: *Schedule is now active.*
Details about the schedule including the frequency and any expiration date are displayed. The **Start Schedule** button is changed to **Stop Schedule**.

6. If you want to pause the schedule run, click **Pause Schedule** at the far right.

![Pause Schedule](image)

7. If you want to resume the schedule run, click **Resume Schedule**. This toggles the button name to **Pause Schedule**.

**Note:** You can also pause a schedule from the Integrations page available under the Monitoring tab. See **Monitor Integration Runs**.

---

**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 Integration page.
2. Find the integration on which the scheduled run is defined.
3. Select **Schedule** from the menu at the right.
   - The Schedule and Future Runs page is displayed.
4. From the menu, click **View Past Runs**.
   - The Track Runs page is displayed. By default, all integration runs are displayed.
5. Filter the display of integrations under **Run State**:
   - Click **Failed** to display integration runs that failed.
   - Click **In Progress** to display any integration runs currently in progress.
   - Click **Completed** to display all completed integration runs.
   - Click the dropdown list to filter the display of runs by 1 hour, 6 hours, 1 day, 2 days, 3 days, or the retention period.
6. If a schedule run has failed (for example, the target FTP server did not have the correct write permissions), click the **Resubmit** icon at the far right to resubmit the schedule run.

If you selected the **Delete Files After Successful Retrieval** checkbox on the Configure File Read page when configuring the inbound FTP Adapter, the file is deleted from the source directory if the file transfer is successful or unsuccessful this time.
This action creates a **RESUBMITTED RUN ID: number** message to the right of the schedule name and original run ID. If you resubmit again, you end up with the following messages to the right of the schedule name:

- **RUN ID: number**: The run ID for the first resubmission, which failed.
- **RESUBMITTED RUN ID: number**: The run ID for the latest submission.
- **ORIGINAL RUN ID: number**: The run ID for the initial submission.

7. Click the **RESUBMITTED RUN ID: number** message link to go to the Tracking page.

8. 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. For this example, the write portion of the integration failed because of a permissions issue.

9. Select **Actions > Audit Trail** to view specific details about the error, such as the target directory not being defined with write permissions.

### View Future Runs

You can view the status of future scheduled integration runs.

To view future runs:

1. Go to the Integration page.
2. Find the integration on which the scheduled run is defined.
3. Select **Schedule** from the menu at the right.
   
   The Schedule and Future Runs page is displayed.

4. Click the menu to display tasks you can perform.
   
   a. Click **Submit Now**.
   
   b. Click **View Past Runs**.

   Details about the future runs are displayed. You can filter to display 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 cancelling. The time zone in which the schedule was created is also displayed.
Edit an Integration Schedule

You can edit a schedule for an integration run.

To edit an integration schedule:

1. Go to the Integration page.
2. Find the integration on which the scheduled run is defined.
3. Select Schedule from the menu at the right.
   The Schedule and Future Runs page is displayed.
4. On the far right, click the Edit icon.
5. Edit the schedule. See Create a 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 Integration page.
2. Find the integration on which the scheduler run is defined.
3. Select Schedule from the menu at the right.
   The Schedule and Future Runs page is displayed.
4. Find the schedule in the list that you want to delete.
5. On the far right, click the Delete icon.

6. Click Yes when prompted to confirm.

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 navigation pane, click Integrations.
3. For the integration schedule that is locked, click the menu at the far right, then select Schedule.
   The Schedule and Future Runs page is displayed.
4. Click the menu in the upper right corner, then click Unlock Schedule.
5. Click Unlock when prompted. Note that unlocking an integration schedule may cause data loss.

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
Monitor Integration Runs

You can pause and resume scheduled runs for an integration from the Monitoring page.

To monitor integration runs:

1. In the navigation pane, click Integrations, then click the < arrow next to Designer.
2. In the navigation pane, click Monitoring.
3. In the navigator, click Integrations.
   For scheduled orchestrated integrations, the calendar icon is displayed.
4. If you want to resubmit a run of an integration, click Submit Now. This button is available only for integrations that have an FTP Adapter as a trigger.
5. If you want to pause the scheduled runs, click the Pause Schedule button.
6. If you want to start a scheduled run, click the **Schedule Not Started** button.

---

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 when you want to test a scheduled integration.

To create ad-hoc integration runs:

1. In the navigation pane, click **Integrations**.
2. Find an integration on which a schedule is defined (indicated by the calendar icon).
3. Click the **Integration Details** icon at the far right.
4. Copy the URL from the **Submit Now Link** 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 at the far right. The Schedule and Future Runs page is displayed.

7. Click the View Past Runs link in the upper right corner. 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.
Creating 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
- Create an Integration to Subscribe to Oracle Integration

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:

1. Follow the steps in Create Integrations to create an integration to publish to Oracle Integration. This creates an integration pattern with a predefined Oracle Integration Messaging invoke connection that enables you to publish messages to Oracle Integration.

2. In the integration designer, drag an adapter from the Connections panel on the right to the trigger (source) area of the canvas. For this example, an Oracle Engagement Cloud 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 Engagement 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 pattern.
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 pattern 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 pattern.

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.

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 the Tracking icon, 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 click the Tracking icon, 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.

### 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 Tracking page and (if integration errors have occurred) the Errors page. If you created scheduled parameters, they are available for assignment as business identifiers.

To assign business identifiers:

1. In the navigation pane, click **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. 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. Select the checkbox if you want to make this the primary business identifier. At least one identifier is required. If you only add one, it is automatically selected as the primary key.
6. 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 Tracking page or (if there is an integration error) the Errors page during runtime.
7. 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 Tracking page to filter messages.
8. Click **Done**.
Delete Business Identifiers

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

To delete business identifiers:
1. In the navigation pane, click **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 Tracking page.

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.
- Oracle-created integrations from the Oracle Marketplace. You import integrations from the Oracle Marketplace as part of a package. These integrations are designated with a **BUILT BY ORACLE** message that is displayed next to the integration name on the Integrations page. You cannot edit these integrations, but you can view their contents, including mappings and business identifiers. You must edit the connections in these integrations to include endpoint credentials relevant to your business requirements. You can also clone these integrations, which enables you to edit the cloned version of the integration.

**Import a User-Created Integration**

To import a user-created integration:
1. In the navigation pane, click **Integrations**.
2. In the upper right corner, click **Import**.
3. Click **Browse** to select the file to import. If you are importing a single integration, select the JAR file to import. If you are importing a package of integrations, select the PAR file to import.

Import and Export Integrations

You can import and export both integrations and lookups to share them between Oracle Integration environments.

See the following topics:
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.

To export an integration:

1. In the navigation pane, click Integrations.
2. Find the integration to export in the Integrations list, click the Actions menu, and then select Export.
3. In the dialog that appears, select Save File, and then click OK.
4. 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.
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 navigation pane, click **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. 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** and **Adapter Configuration Reference**.

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**

- 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 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.

To regenerate a WSDL file for integrations:

1. In the navigation pane, click **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 Artifact**.

You are prompted with a message indicating that WSDL regeneration impacts the mappings in the integration.

b. Click **Yes**.

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 **Actions** at the top of the page.
   
a. Click **Regenerate endpoints**.

You are prompted with a message indicating that WSDL regeneration impacts the mappings in the integration.

b. Click **Yes**.

This regenerates the WSDLs and any dependent artifacts with the same behavior as described in Step 4.

See Mapping Data of *Using the Oracle Mapper*.
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
• Manage Lookups

Map Data

Use the mapper to drag fields from the source structure to the target structure to map elements between the two.

Topics:
• Create Mappings
• Modify Mappings
• Delete All Mappings
• Map Faults
• Encode and Decode File Attachment Content
• Add Customized Mappings to Prebuilt Integrations
• Remove Customized Mappings from Prebuilt Integrations
• Import Map Files

See Map Data of Using the Oracle Mapper.

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 pattern you are using.

To create mappings in App Driven Orchestration integrations and Scheduled Orchestration integrations:
As you add triggers and invokes to an App Driven Orchestration integrations, a map icon is automatically added. You can also add ad-hoc mappings to this type of integration, such as adding a mapper to a switch action.

1. Click an existing mapper icon or drag a mapper into your integration from the Actions panel to the appropriate location in your integration.

2. Click Edit.
If you click the View icon, note the following details:

- You cannot add or edit mappings.
- You cannot validate mappings.
- You cannot save or erase the XPath expression in the Expression Builder.
- You cannot create or delete elements or mappings in the target context menus.
- You cannot drag source element nodes to target element nodes.
- You can view XSLT code and test your mappings.


To create mappings in Basic Routing integrations:

1. In the middle of the integration, click the Mapper icon for the request, response, or fault map to edit.
2. Click Edit.

See Creating Integrations.

Modify Mappings

Once you create a mapping in an integration, you can return to the mapping and make any necessary changes to how you mapped your data. The integration in which you want to edit the mappings cannot be active.

To modify a data mapping:

1. In the middle of the integration, click the Mapper icon for the request, response, or fault map to edit.
2. Click Edit to invoke the mapper.
3. Make appropriate updates to the mappings.
4. When complete, click Close, then click Apply to save your changes.

See Mapping Data of Using the Oracle Mapper.
Delete All Mappings

You can delete all mappings in the mapper. This action deletes all source-to-target mappings in the mapper and all mapper statements created in the Mapping Builder.

1. Click the Mapper icon in the middle of the integration for the map to delete. For this example, the request mapper is selected, but you can also delete all mappings in another mapper, such as the response mapper, or any request or response enrichment mapping you created.

2. Click Delete.

3. Click Yes when prompted to confirm.

The green shading is removed from the mapper, indicating that the mapper is now empty.

See Mapping Data of *Using the Oracle Mapper*.

Map Faults

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*.

### 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 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 Mapping Components - 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.

### Add Customized Mappings to Prebuilt 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*.

### Remove Customized Mappings from Prebuilt 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 integrations:

1. In the navigation pane, click **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:
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.

**Import Map Files**

Review the following topics to learn how to import map files into Oracle JDeveloper and Oracle Integration.

**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.

**Note:**

You cannot view or edit a map file imported into the mapper.

1. In the navigation pane, click **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
- Add Adapters or Domain Names to a Lookup
- Clone a Lookup
- Delete a Lookup
- Update Lookups in Active Integrations
- Export a Lookup
- Import a Lookup
- Create the lookupValue Function

**Create a Lookup**

Create a lookup to map values between applications.

1. In the navigation pane, click **Integrations**, then click **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:
• **Select 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. On the Lookups page, select **Clone** from the menu.
   
   The Clone Lookup dialog is displayed.

2. Enter the lookup information.

3. Click **Clone**.

4. Click **Edit** to further configure your cloned connection.
Delete a Lookup

You can delete a connection from the Lookup menu.

1. Select Delete from the menu.
   The Delete Lookup dialog is displayed.
2. 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 take effect immediately. There is no need to re-activate integrations using a changed lookup for the new lookup value to take effect.

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 Integrations, then click Lookups.
2. Select the Actions menu to the right in the lookup’s row, then select Export to CVS.
3. In the dialog that appears, select Save File, and then click OK.
4. 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:

<table>
<thead>
<tr>
<th>Adapter Name</th>
<th>Identifier to Use in the Import File</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle Eloqua Cloud</td>
<td>eloqua</td>
</tr>
<tr>
<td>Oracle Engagement Cloud</td>
<td>osc</td>
</tr>
<tr>
<td>Oracle Messaging Cloud Service</td>
<td>oms</td>
</tr>
<tr>
<td>Oracle Service Cloud</td>
<td>rightnow</td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th>Adapter Name</th>
<th>Identifier to Use in the Import File</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle HCM Cloud</td>
<td>hcm</td>
</tr>
<tr>
<td>Oracle ERP Cloud</td>
<td>erp</td>
</tr>
<tr>
<td>Salesforce Cloud</td>
<td>salesforce</td>
</tr>
</tbody>
</table>

To import a lookup:

1. Locate the CSV file containing the lookup table you want to import.
2. In the navigation pane, click **Integrations**, then click **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
- Select the Lookup Table
- Select the Source and Target Columns
- Specify the Default Value
- Review Your Lookup Table Selections

**Access the Build Lookup Function Wizard**

The Build Lookup Function wizard for creating the `lookupValue` function parameter values is accessible from the Expression Builder in Oracle Integration.

To access the Build Lookup Function wizard:

```markdown
Note:

You must already have created lookups to use this wizard. See Manage Lookups.
```
1. Go to the Integrations page.
2. Open an integration.
3. Invoke the Expression Builder from within the integration. For example:
   - In an orchestrated integration, edit a switch activity.
   - In a basic integration, click the Filter link.
   The Expression Builder is displayed.
4. Expand Components.
5. Expand Functions > ICS.
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.

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lookup Table</td>
<td>Select the lookup table to use in the function.</td>
</tr>
<tr>
<td></td>
<td>You can view the lookup description by clicking the information icon in the</td>
</tr>
<tr>
<td></td>
<td>table. This can guide you in selecting the required lookup table. The number of columns defined in the lookup is also displayed.</td>
</tr>
</tbody>
</table>

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.
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.

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.

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.
<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resulting Expression</td>
<td>Displays the expression you defined on the previous pages of the wizard. The <code>lookupValue</code> function takes the following format:</td>
</tr>
<tr>
<td></td>
<td><code>lookupValue(dvmLocation, srcColumn, srcValue, targetColumn, defaultValue)</code></td>
</tr>
</tbody>
</table>

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:

```plaintext
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.
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 exchanges between your on-premises applications and Oracle Integration.

Topics:
- Create an Agent Group
- Download and Run the Connectivity Agent Installer
- Create a Connection with an Agent Group
- Upgrade the Connectivity Agent Automatically
- Back Up and Recover the Connectivity Agent
- Delete an Agent Group
- Agent Behavior in a Decommissioned Instance or HTTP 404/401 Error Response Codes
- Troubleshoot Connectivity Agent Issues

See About Connectivity Agents and Integrations Between On-Premises Applications and Oracle Integration.

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. Only one connectivity agent 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 navigation pane, click Integrations, then click Agents.
2. Click Create Agent Group.
   The New Agent Group — Information dialog is displayed.
3. Enter the following information, then click Create.
### Agent Group 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.

### Agent Type

**Connectivity Agent** 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.

---

### Download and Run the Connectivity Agent Installer

You must download and run the connectivity agent installer to install the agent in your local environment.

### System Requirements

Satisfy the following system requirements before installing the on-premises connectivity agent.

- The agent is certified with Oracle JDK Version 8. The JDK installation can be shared with other products installed on the same host. However, ensure that the JDK installation is not modified for use with these other products.
- The agent is certified on the following operating systems:
  - Oracle Enterprise Linux 6.x
  - Oracle Enterprise Linux 7.2
  - Oracle Enterprise Linux 7.5
  - RedHat Enterprise Linux 6.6
  - RedHat Enterprise Linux 7.2
  - RedHat Enterprise Linux 7.5
  - Suse Linux Enterprise Edition 12 SP2
Note:
If you install the agent on Windows, automatic upgrade of the agent is not supported. You must manually upgrade the agent after your instance is upgraded to the latest version. See Running the Connectivity Agent on a Windows Host Requires a Manual Upgrade.

Note:
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.

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. Restart the agent.

   `java -jar connectivityagent.jar`

   You can restart the agent as a background process. See Run the Connectivity Agent Installer as a Background Process.

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
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 ServiceAdministrator or ServiceDeveloper role to download the agent. See Grant Access and Secure of Administering Oracle Integration.

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 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:**

```text
# 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:

- **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 username. 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 username at agent startup and it is not persisted.
- **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.
- **Proxy Parameters**: These parameters are only required if the connectivity agent is used with a proxy in the on-premises environment.

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:

   ```csh
   setenv PATH = $JAVA_HOME/bin:$PATH
   ```

10. Run the connectivity agent installer from the command prompt:

    ```java
    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 :

   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 or otherwise invalid 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:

   ```bash
   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`.
- `-alias alias_name`: Any name to uniquely identify the imported certificate in
  the keystore.
- `-file certificate_file`: Absolute path of the certificate file.
Use the Agent in High Availability Environments

You can use the connectivity agent in high availability environments. You install the connectivity agent twice on different hosts. There are no differences in agent installation. You specify the same agent group identifier during both installations.

Note the following details:

- The File Adapter is not supported in high availability environments. When using the File Adapter and some groups have multiple instances, use a dedicated agent group (with one agent only).
- Any new agent group created is enabled for high availability. Existing agent groups created prior to high availability being supported have no high availability capabilities and cannot be used.
- 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.
- 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).
- Both hosts on which the agent is installed must have the same network setup.
- To enable an older agent integration for high availability, you must create a new group, update the adapter connections in the integration to refer to the new group, and re-activate the integration.
- 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.

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.

![Agents Page](image)

5. Go to the Monitor Agents page and note the same information. In this case, both agent instances are running.

![Monitor Agents Page](image)

Run the Connectivity Agent Installer as a Background Process

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:

1. Update InstallerProfile.cfg with oic_USER and oic_PASSWORD values and then use `nohup` to run the agent process.

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.
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 **Connections**.
2. Click **Create**. See **Create a Connection**.
3. Select the adapter to configure as a target endpoint. See **About Connectivity Agents and Integrations Between On-Premises Applications and Oracle Integration**.
4. Configure the connection properties and security.
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 **Close**.
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 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 \texttt{agent\_home} directory of your installation.
7. The endpoints are quiesced.
8. The agent is shut down and restarted.
You are notified of upgrade success.

\textbf{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.

\section*{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.

\textbf{Note:}

- Perform regular backups of the \texttt{agent} directory located under \texttt{AGENT\_INSTALL\_DIR/agent\_home}.
- 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. Go to the Agents page in Oracle Integration.
2. Click \textbf{Download} > \textbf{Connectivity Agent}.
3. Save the \texttt{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.

---

### 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. On the Oracle Integration home page, click **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. From the menu, select **Delete**.
5. Select **Yes** 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**.

**Troubleshoot Connectivity Agent Issues**

This section describes how to troubleshoot and resolve connectivity agent issues.

- **Java Memory Errors**
- **Authorization Error When Restarting the Agent**
- **Run the Connectivity Agent on a Windows Host Requires a Manual Upgrade**
- **Add or Change the Non-Proxy Host Configuration After Installation in the CpiAgent.properties File**
- **Class Loading Conflict When Sharing the JDK Instance with Another Product**
- **Connectivity Agent Log File Location**
- **Error When Using the Connectivity Agent in an Oracle Integration Classic (User-Managed) Environment**
- **Agent Behavior in a Decommissioned Instance or HTTP 404/401 Error Response Codes**
- **Verify Endpoint Accessibility when Agent is Installed with a Proxy Host**
- **Unlock the Agent Group**
- **Failure to Send a Response Due to java.net.SocketException: Connection Reset Error**
- **Agent Installation on Linux Fails When Using an Installer Copied Using winscp**
- **Agent Performance Tuning**
- **Integration Activation Error Due to Change in JDK Location**
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 My Services 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.

Run the Connectivity Agent on a Windows Host Requires a Manual Upgrade

To manually upgrade the agent when an upgrade is required.

1. Stop the connectivity agent.
2. Download the agent ZIP file from Oracle Integration.
3. Back up the lib folder in agenthome.
4. Back up the version in agenthome.
5. Replace the lib folder in agenthome with the one from the downloaded ZIP file.
6. Replace version in agenthome with the one from the downloaded ZIP file.

7. Replace connectivityagent.jar in the AGENTINSTALL directory with the one from the downloaded ZIP file.

8. Restart the connectivity agent.

   ```
   JDK_HOME/bin/java -DenableAutoUpgrade=false -jar connectivityagent.jar
   ```

9. Update the agent version. Replace the place holders with actual values before running this command:

   ```
   ```

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 InstallerConfig.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 agenthomes/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.)

   ```sql
   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.


   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:

```text
com.sun.jersey.api.client.ClientHandlerException: java.net.SocketException: Connection reset
at com.sun.jersey.client.urlconnection.URLConnectionClientHandler.handle(URLConnectionClientHandler.java:155)
at com.sun.jersey.api.client.Client.handle(Client.java:652)
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.java:161)
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.java:a:80)
Integrate with Processes and Services

You can integrate Oracle Integration with processes and other services.

Topics:
• Create an Integration that Invokes a Process
• Connect to Oracle Mobile Cloud Service
• Connect to Oracle API Platform Cloud Service

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:
1. In the navigation pane, click Integrations.
2. In the upper right, click Create.
3. Select Orchestration.
   The Create New Integration dialog is displayed.
4. Complete the fields to create an orchestrated integrations, then click Create.
5. On the right side, click Integration Artifacts.
7. Drag the process to the Plus sign in the integration.
   The Select Process wizard is displayed.
8. On the Basic Info page, enter a name and optional description, then click Next.
9. 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 field, select the process name. The process must already be activated and must be a 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. Click Next.
10. 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.

11. Complete the design and save the integration.
12. 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 navigation pane, click **Integrations**, then click the < arrow next to **Designer**.
2. Click **Settings**, then 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 you want 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.
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 an Integration
• Deactivate an Integration
• Manage Packages
• Modify an Integration
• Edit the Endpoint Information in an Integration
• Enable or Disable Tracing on Active Integrations
• View the Trigger, Invoke, and Enrichment Details of an Integration
• Clone an Integration
• Create a Draft of an Integration
• Delete an Integration
• Unlock an Integration
• Cancel a Running Integration Instance
• Download Generated Artifacts for an Integration
• View the Contents of an Incomplete Integration
• Edit an Integration with Incomplete Connections
• Reactivate Integrations After Instance Upgrade to View the Latest Business Identifier Tracking Behavior
• Filter the Display of Integrations By Type
• Change the Time Zone
• Use Libraries to Manage Functions for Integrations
• Manage an Integration as an API with Oracle API Platform Cloud Service

Activate an Integration

Once you create an integration and the progress indicator shows 100 percent, 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.

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 navigation pane, click **Integrations**.
2. In the Integrations list, locate the integration you want to activate and go to the far right end.
3. Click the icon to activate the integration.

The Confirmation dialog is displayed.

4. Select options appropriate to your integration.

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contribute integration mappings to Oracle Recommendations</td>
<td>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 <strong>Settings &gt; Recommendations</strong> in the navigation pane.</td>
</tr>
<tr>
<td>Enable tracing</td>
<td>Click to enable detailed tracing information in the activity stream. 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 <strong>Enable tracing</strong> checkbox. If you do not enable tracing, minimal logging details are created in ics-flow.log to indicate when the activated integration begins and completes execution (regardless of execution success or failure). You can download the logs on the Dashboard page.</td>
</tr>
</tbody>
</table>
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 [Create a Schedule](#).

   A status message is displayed in the banner at the top of the page. For example:

   ![Integration Echo (1.2.0) submitted for activation. Click refresh if status is in progress](image)

   - Upon successful activation, click on "How To Run" to get endpoint URL to trigger this integration.
   - You can also go to Tracking page to track instances.

6. If integration is unsuccessful, a yellow **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.

   ![Activation Failed](image)

   - 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.

7. If activation is successful, the status of the integration changes to **ACTIVE** in the row. Immediately to the right of the **ACTIVE** icon is an icon that when clicked shows the submit now URL and Tracking Runs page link for scheduled
integrations and the endpoint trigger URL and Tracking page link for unscheduled integrations:

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 Tracking page, the menu includes an option called View Activity Stream for viewing payload details.

To access the detailed trace logging information:

a. In the navigation pane, click the < arrow next to Designer.

b. Click Monitoring, then click Dashboards.

c. 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 at the far right of the integration. The active integration is displayed with a message saying Viewing.

Note the following details about read-only mode:

• No Save button and Actions button are displayed.
• There is no Connections Palette for adding adapters.
• You can click through multiple parts of the integration to view configuration details, such viewing the business identifiers under the Tracking tab, 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.

Deactivate an Integration

You can deactivate an integration to stop it from processing any new messages. If you want to modify an active integration, you need to deactivate it first.

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 navigation pane, click Integrations.

2. On the Integrations page, find the integration you want to deactivate.
To view only active integrations, select Active in the Integrations list. You can also filter by integration name or integration type (prebuilt, custom, or developed) to narrow down the list.

3. In the row containing the integration you want to deactivate, click the Active icon to deactivate the integration.

4. Click Deactivate on the dialog that appears. A deactivation progress bar is displayed at the top of the dialog.

If the integration 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.

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:
- View the Integrations in a Package
- Import a Package
- Export a Package
- Update a Package
- Delete a Package

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 navigation pane, click Integrations, then click Packages.

2. Click the name of the package or select View Integrations from the menu at the far right.

   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 **Integrations**, then click **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 **Integrations**, then click **Packages**.
2. From the menu at the far right, select **Export**.
3. 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. From the menu, select **Edit**.

The integration is displayed.

3. From the menu, select **Primary Info**.
4. In the **Name** 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.

5. Click **Save**, then click **Close**.

6. In the navigation pane, click **Packages**.

7. 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 **Integrations**, then click **Packages**.

2. From the menu at the far right, select **Delete**.

3. Click **Yes** 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**.

### 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 navigation pane, click **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 **Search** field or filter integrations by selecting an option from the **Filter By** list. From this list, you can filter by **Type** (Customized, Developed, and Prebuilt), **Pattern** (Schedule, File Schedule, Map Data, Notification, Publish to ICS, Subscribe To ICS, and Undefined), **Status** (Draft, Configured, Active, or Failed), or **Style** (Template or Orchestration). Search or filter criteria are displayed in the banner above the returned list of integrations. To remove search or filter criteria, click the **x** icon in the banner.

3. On the row that contains the integration you want to change, click the **menu**, and select **Edit**.

4. To modify the source or target configuration, click the connection on the canvas and click **Edit** on the menu that is displayed.

5. Modify any of the open fields in the wizard that appears. See **Connection Configuration Reference** for instructions.
6. To assign a new connection as the source or target, click the connection to delete, then click **Delete** on the menu that is displayed.

7. From the right side, drag the new adapter from the panel to the connection on the canvas that you want to replace. Configure the new connection. See **Connection Configuration Reference** for instructions.

8. To modify a data mapping, click the appropriate map icon and update the mappings. See Mapping Data of Using the Oracle Mapper.

9. When you are done making changes, click **Save** and then click **Close**.

---

**Note:**

If you have an integration that is locked and stuck in Draft mode, wait for the lock to expire. Expiration typically occurs within 30 minutes. This situation can occur if your browser crashes while editing an integration.

---

### Edit 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 navigation pane, click **Integrations**.
2. Select the integration to edit.
3. In the integration, select the trigger or invoke endpoint to edit, then click the **Edit** icon.
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.

### 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.

1. In the navigation pane, click **Integrations**, then click the < sign next to **Designer**.
2. Click **Settings**, then click **Tracing**.
3. Select an option below to configure the tracing level.

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integration Level</td>
<td>Select to enable or disable tracing at the individual integration level on the Integrations page. This enables you to select <strong>Tracing</strong> 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.</td>
</tr>
<tr>
<td>Element</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Global Tracing On</td>
<td>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 beneath the activate icon for all integrations on the Integrations page.</td>
</tr>
<tr>
<td></td>
<td><strong>TRACE WITH PAYLOAD</strong></td>
</tr>
<tr>
<td></td>
<td>Tracing details are displayed in the activity stream.</td>
</tr>
<tr>
<td></td>
<td>If you attempt to disable tracing for an individual integration on the Integrations page by selecting <strong>Tracing</strong> 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 <strong>Integration Level</strong> to enable or disable tracing at the integration level.</td>
</tr>
<tr>
<td>Global Tracing Off</td>
<td>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 <strong>Tracing</strong> 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 <strong>Integration Level</strong> to enable or disable tracing at the integration level.</td>
</tr>
</tbody>
</table>

**View the Trigger, Invoke, and Enrichment Details of an Integration**

You can view the details associated with trigger, invoke, and enrichment endpoints of an integration.

To view the trigger, invoke, and enrichment details of an integration:

1. In the navigation pane, click **Integrations**.
2. On the Integrations page, find the integration you want to view. You can filter the display of integrations by their current status on the left side of the page.
3. Click the trigger, invoke, or (if configured) enrichment endpoint of the integration.
4. Click the **View** icon.
5. 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 navigation pane, click Integrations.
2. On the Integrations page, find the integration you want to clone. You can filter the display of integrations by their current status on the left side of the page.
3. In the row containing the integration you want to clone, click the icon on the far right, then click Clone.
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 navigation pane, click Integrations.
2. On the Integrations page, find the integration for which you want to create a draft. You can filter the display of integrations by their current status on the right side of the page.

3. In the row containing the integration for which you want to create a draft, click the menu, and select Create Version.

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 Integrations.

2. On the Integrations page, find the integration you want to delete. You can filter the display of integrations by their current status on the left side of the page.

3. In the row containing the integration you want to delete, click the menu on the far right, then click Delete.

4. Click Yes on the dialog that appears.

Unlock an Integration

When an integration is in edit mode and the browser crashes, the integration becomes locked, which prevents it from being edited. You can unlock resources such as integrations that you locked. Users with the Administrators role can also unlock resources locked by any user.

To unlock the integration:

1. Log in to Oracle Integration.

2. In the navigation pane, click Integrations.

3. For the integration that is locked, click the menu at the far right, then select Unlock.
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 navigation pane, click **Integrations**, then click the < arrow next to **Designer**.
2. Click **Monitoring**, then click **Tracking**.
3. Find the integration to cancel.

4. In the row containing the integration you want to cancel, click the menu on the far right, then click **Discard**.

Scheduled instances cannot be canceled from the Track Instances page. Selecting **Discard** for a scheduled instance causes a message to be displayed above the banner with a link to the Schedule Overview page for canceling the running instance.

Download Generated Artifacts for an Integration

You can download artifacts of an integration as a zip file for viewing and analysis. You can download completed integrations, locked integrations, and active integrations, but not integrations that are in a draft state. The artifacts of active integrations are regenerated when downloaded. However, this action does not fetch the previously deployed artifacts from runtime. Instead, it regenerates and provides artifacts for download only (has no impact on the deployed artifacts). Therefore, these artifacts may not be fully identical with the previously deployed artifacts.

Downloading artifacts differs from exporting an integration. Exporting creates an IAR file that consists of the design time metadata of the integration. The generated artifacts downloaded are the same JAR files that are deployed to runtime through the activation process. Therefore, they are generated from the design time metadata and also contain the runtime metadata.

To download generated artifacts for an integration:

1. In the navigation pane, click **Integrations**.
2. On the Integrations page, find the integration for which to download artifacts.
3. In the row containing the integration, click the menu on the far right, and click **Download Artifacts**.
4. Specify a directory in which to save the zip file when prompted.

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 navigation pane, click **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:
   1. In the navigation pane, click **Integrations**.
   2. 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.

**Reactivate Integrations After Instance Upgrade to View the Latest Business Identifier Tracking Behavior**

After your Oracle Integration instance is upgraded to a newer version, you must reactivate your integrations to view the latest and correct business identifier behavior in the Tracking page.
Filter the Display of Integrations By Type

You can filter the display of integrations by their type (custom, user-developed, or prebuilt).

To filter the display of integrations by type:
1. In the navigation pane, click **Integrations**.
2. Select the **Filter by** list to filter the display of integrations.
3. Select the **Sort By** list to filter the display of integrations by the last update or name.

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 Settings** 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 Tracking page).

Use Libraries to Manage Functions for Integrations

JavaScript libraries provide a means to register and organize JavaScript for use in integrations. Library functions are automatically available for you to drag from the **Actions** palette to your orchestration integrations.

You can use the library functionality to:
- Register a new JavaScript library
- See a list of all registered 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 registered JavaScript library
- Download a library source file to your desktop
- Add library functions to the orchestration **Actions** palette.

Topics:
- Register a JavaScript Library
Register a JavaScript Library

In order for custom function calls to appear in the integration Actions palette, you must register the libraries that contain them.

1. In the navigation pane, click Integrations, then click Libraries.
2. Click the Register button in the title bar.
3. In the Register Library dialog box, click Browse to select a JavaScript (.js) library file.
4. Specify a name, identifier, version number and optional description.

**Example 7-1  Code Format**

Function return parameters must be named like the example below. You should configure the return parameter type in the metadata UI only if the return parameters are named.

Consider the following code:

```javascript
function add ( param1, param2 )
{    return param1 + param2;
}
```

Although the above syntax is valid, the return parameter will not appear in the metadata UI 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:

```javascript
function add ( param1, param2 ) {
    var retValue = param1 + param2;
    return retValue;
}
```

**Example 7-2  Function Within Another Function**

Functions defined within another function do not appear in the metadata UI for configuration.

Consider the following code:

```javascript
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. Thus, foo does not appear in the metadata UI for configuration.

**View Library Details**

View details about a library without changing any information.

1. In the navigation pane, click **Integrations**, then click **Libraries**.

2. Select **View** from the menu at the end of the row on which the library is listed.

   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.

   You can use the menu on this page to view primary information about the library and to export XML metadata to your desktop.

3. When you are done, click **Close** on the title bar.

**Edit Library Details**

Edit details about a library.

1. In the navigation pane, click **Integrations**, then click **Libraries**.

2. Either click on the library name or select **Edit** from the menu at the end of the row on which the library is listed.

   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 are presented with the following options:

   - You can set the classification type to be **Orchestration**, enabling the function to be used in orchestrations.
   - 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
- Export XML metadata to your desktop
- Delete the library
- Save your changes (the same as clicking Save on the title bar)

3. When you are done, click **Save** and **Close** on the title bar.

### 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 **Integrations**, then click **Libraries**.

2. Click the number in **Used by Integrations** label in the line beneath the library’s name. The number indicates the number of integrations that use functions in the library.

When you click the number, the Library Is In Use popup is invoked. It lists the names of the integrations that use the library's functions. This popup 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 navigation pane, click **Integrations**, then click **Libraries**.

2. Select **Delete** from the menu at the end of the row on which the library is listed.

If any of the library's functions are being used by an integration, the Library Is In Use popup is invoked and you are prevented from deleting the library. You must remove the functions from the integration before you delete it.

### 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 **Integrations**, then click **Libraries**.

2. Select **Download Library File** from the menu at the end of the row on which the library is listed.
Use Library Functions in Orchestrations

Library functions are available from the orchestration Actions palette from where you can incorporate them into your orchestrations.

After you register and edit your library functions, they are available from the orchestration Actions palette. You can drag them from the Actions palette to your orchestrations. See Add a JavaScript Action.

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
• Group an Integration in an API in Oracle API Platform Cloud Service
• Ungroup an Integration from an API

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 Integrations.
2. In the row containing the integration you want to manage with API Platform Cloud Service, click the Actions icon on the far right and then click Manage API with API P CS.
3. From the Operation list, select Create New API.
4. In the API Name field, enter a name for the API.
5. In the **Version** field, enter a version for the API.

6. 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.

7. **(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.

8. **(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.

9. **(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.

10. 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 in Using Oracle API Platform Cloud Service.

To group an integration in an API:

1. In the navigation pane, click **Integrations**.
2. In the row containing the integration you want to manage with API Platform Cloud Service, click the **Actions** icon on the far right and then click **Manage API with APIP CS**.

3. From the **Operation** list, select **Add to Existing API**.

4. In the **API Name** field, enter the name of the API you want to add the integration to.

   **Tip:**

   Click the **Search: API Name** icon to view the APIs you can add the integration to.

5. **(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.

6. **(Optional)** To publish the API to the Developer Portal, select **Publish API**, and then enter a vanity name into the **API Portal URL** field. This option is grayed out if the user doesn’t have the Manage API grant for the API you selected.

7. 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.
To ungroup an integration from an API:

1. In the navigation pane, click **Integrations**.

2. In the row containing the integration you want to ungroup from an API, click the **Actions** icon on the far right and then click **Manage API with APIP CS**.

   The Manage API with APIP CS dialog appears.

3. In the **Operation** list, ensure that **Remove from Existing API** is selected, and 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.
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
- Manage Errors
- Manage Business Identifiers for Tracking Fields in Messages
- View Preinstalled Adapters
- Send Service Failure Alerts, System Status Reports, and Integration Error Reports by Notification Emails
- Set Logging Levels and Download Diagnostic Logs
- Report Incidents
- Purge and Retain Data in the Database

**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
- View System Health
- View Design-Time Metrics
- Monitor Integrations
- Monitor Agents
- Monitor Scheduled Integration Runs

**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 **Integrations**, then click the < arrow next to **Designer**.

2. Click **Monitoring**, then click **Dashboards**.

The Dashboard page provides the following details:

- **Success rate for messages (acceptable, adverse, or severe).** Click the percentage value or wording to access the Errors by Integration page.

- **Number of initiated messages in the last three days.** Click the value or wording to access the Tracking page.

- **Number of failed messages in the last three days.** Click the value or wording to access the Error Message Details page.

- **Number and percentage of currently used connections.** Click the value, percentage, or wording beneath the percentage to access the Errors by Connection page.

- **Number and percentage of currently active integrations.** Click the value, percentage, or wording beneath the percentage to access the Errors by Integration page.

- **Number and percentage of scheduled integrations, including those currently paused.** Click the value, percentage, or wording beneath the percentage to access the Integrations page listing all scheduled integrations.

- **Activity streams link.** Click the link to view the activity stream.

- **Download Logs menu.** Click the link to show options for downloading the Oracle Integration logs or diagnostics 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.

- **Download incidents link.** Click to download the created incident report. See **Report Incidents**.

- **Graphs showing the hourly and daily history of total, successful, and failed messages.** You can place your cursor over the bars in each graph to display the total number of successful messages and failed messages.

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.
View the Activity Stream

You can view the activity stream and message activity of invoked integrations.

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.

- The supported size of the activity stream of an integration is restricted to less than 10 MB. If the activity stream is greater than 10 MB, only 200 records can be viewed.

To view the activity stream:

1. On the Dashboard page, click the Activity Stream link.

The Activity Stream page shows details about the activity stream of invoked integrations.
2. Click **Download** to download the activity stream.

3. From the **Activity Stream** dropdown list, select **Integrations** to show the message activity (received, processed, successful, and failed) of integrations.

**View System Health**

You can view information about the system health of Oracle Integration components. As you navigate around Oracle Integration, you receive a system health state that is not older than five minutes. The only exception is the Messaging Service state, which may take longer than five minutes to update.

The state of system health is viewable from the **System Health** selection of the **Integration Health** menu on the Dashboards page.

To view system health:

1. In the navigation pane, click **Integrations**, then click the `<` arrow next to **Designer**.
2. Click **Monitoring**, then click **Dashboards**.
3. From the **Integration Health** menu, select **System Health**.
4. View the current status of system services upon which Oracle Integration relies. A green icon indicates that the service is running as expected. A red icon indicates that errors are occurring that can impact your active integrations. An agent can have three states: available (denoted by a green icon), unavailable (denoted by a red icon), and not configured (denoted by a gray icon). When database space usage reaches the value you specified on the Database Configuration page, the color changes to orange in the **RECENT HEALTH CHECKS** section.
5. View the following details about the database:

- The unquiesced state of the database. This means that users can perform the usual Oracle Integration tasks such as creating and updating integrations, connections, lookups; importing and exporting integrations and packages; activating integrations; and so on. For this example, only 1% of the available database space has been used. When database usage reaches the warning threshold value set on the Database Configuration page, you can receive an email notification if you subscribed to receive system failure notifications. When database space usage reaches the quiesced mode threshold value set on the Database Configuration page, the mode changes to a quiesced state, a red warning sign is displayed, and the color changes to red. When the database enters into a quiesced state, there is a ten minute waiting period after which the runtime does not accept any payloads. You can also receive an email notification if you subscribed to receive system failure notifications when the database enters into a quiesced state. See Send Service Failure Alerts, System Status Reports, and Integration Error Reports by Notification Emails.

- The state of the database purge and the data retention period. The current data retention periods for successful and faulted instances are also displayed. Click anywhere in the DATABASE section to access the Database Configuration page to make changes. See Purge and Retain Data in the Database.

- A link to the purge log in the upper right corner. You can also scroll through a list of the latest purge actions.

- Process data is also reflected in these database space usage values. You can archive and purge Process data. See Archiving and Purging Data of Using Processes in Oracle Integration.

6. View the maximum mount usage in the file system. The warning threshold is 90%, at which time you can receive an email notification if you have configured a notification email address. The color changes from green to orange. See Send Service Failure Alerts, System Status Reports, and Integration Error Reports by Notification Emails. When usage reaches 95%, the color changes from orange to red and Oracle Integration enters into a read-only state, as indicated by a
message at the top of the Integrations page. During this state, you cannot perform tasks such as the following until you have reduced file system usage below 95%:

- Activating integrations
- Importing and exporting integrations and packages
- Creating or updating any resource (for example, lookups, integrations, connections, and so on)
- Downloading logs

- Click the information icon or anywhere in the **FILE SYSTEM** section to display the domains on the POD instance server, the mount points, and the disk space usage of each mount point. The maximum usage of any one of the mounts is displayed. For example, if ICS_Data is 30% and ICS_Tmp is 70%, 70% is displayed.

These numbers represent the used space of the mount points, and not the folder usage. A mount point (like a drive) has used space and total space. A folder does not normally have free space.

**View Design-Time Metrics**

You can view the design-time metrics of Oracle Integration, including details about packages, integrations, connections, lookups, and agents.

To view the design-time metrics of Oracle Integration:

1. In the navigation pane, click **Integrations**, then click the `<` arrow next to **Designer**.
2. Click **Monitoring**, then click **Dashboards**.
3. From the **Integration Health** menu, select **Design-Time Metrics**.
4. View details about the following components in Oracle Integration:
   - The number of developed and prebuilt packages.
   - The state of integrations (draft (under construction), configured, active, and failed).
   - The state of connections (draft, configured, and in use).
   - The number of lookups.
   - 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
   - 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).

8. From the **Actions** menu, select **Download** to download the design-time audit trail.

**Monitor Integrations**

On the Oracle Integration dashboard, you can see how your running integrations are processing messages, such as how many messages have been received and processed, how many successful messages and errors have occurred, and the overall success rate. Only activated integrations are listed on this page.

To monitor integrations:

1. In the navigation pane, click **Integrations**, then click the < arrow next to **Designer**.
2. Click **Monitoring**.
3. In the navigation pane, click **Integrations**.

A list of running integrations appears, along with processing information about the number of messages received, the number of messages processed, the number of successful messages, and the number of failed messages. Up to 80 activated integrations can be displayed. If the integration you are looking for is not displayed
among the 80, use the search facility. For scheduled integrations, navigation links are provided that take you to the Schedule and Future Runs page.

4. From the dropdown list, select the time period or retention period during which to search for integrations.

You can also specify a custom time range with which 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, Track Runs, or Errors page. See Step 3 of Track Business Identifiers in Integrations During Runtime for details.

5. From the Integrations dropdown list, select the Activity Stream to view the activity stream for the integrations.

Monitor Agents

You can monitor the agent groups and their associated on-premises agents in Oracle Integration.

To monitor an agent:

1. In the navigation pane of Oracle Integration, click < to the left of Designer.
2. Click Monitoring, then click Agents.
3. The Agent Monitoring page shows details such as the time at which the agents were last updated and the on-premises 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.
4. Click the number above AGENTS or click the agent group name to see the on-premises agent associated with this group.
5. At the far right, click the information icon to display details about the agent group such as creation date and the last startup date of the on-premises 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).

### Monitor Scheduled Integration Runs

You can monitor scheduled integration runs on the Track Runs page. You can view the total number of messages (records) processed, the number of successfully processed messages, and the number of failed messages. Scheduled integrations can be those with a defined schedule or those submitted ad-hoc through the **Submit Now** option.

To monitor scheduled integration runs:

1. Access the Track Runs page in either of two ways:
   
a. Access the Track Runs page directory from the navigation pane by clicking **Integrations**, then clicking the `<` arrow next to **Designer**.
   
b. Click **Monitoring**, then click **Runs**.

   The Track Runs page is displayed.

   or

   a. Access the status of a scheduled integration instance submitted through the **Submit Now** option by going to the Integrations page.
   
b. Find the scheduled integration that you want to run.

   c. From the menu at the far right, select **Submit Now**.

   If the submittal was successful, the following message is displayed.

   ```
   Submit Now request to run integration integration_name (version_number) was submitted successfully with request id number.
   ```

   d. Click the ID number.

   The Track Runs page is displayed. If the submitted integration does not appear, click the **Refresh** icon. Details about the run are displayed.

2. From the dropdown list, select the time period or retention period during which to search for instance runs.

   You can also specify a custom time range with which 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, Integrations, or Errors page. See Step 3 of **Track Business Identifiers in Integrations During Runtime** for details.
3. Click to view the tracking status of the scheduled integration on the Tracking Instances page or click to rerun the scheduled integration.

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 audit trail, business identifiers and values, and message payload in failed instances.

Topics:
• View Specific Error Details
• Resubmit Failed Messages
• Discard Errors
• View the Status of Message Recovery
• View the Audit Trail of Failed Integration Instances
• View Business Identifiers in Failed Integration Instances
• View the Message Payload of a Failed Integration Instance

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 navigation pane, click Integrations, then click the < arrow next to Designer.
2. Click Monitoring, then click Errors.
3. Filter the display of errors by integration, connection, or recovery job, as needed.
4. From the dropdown list to the right, select the time period or retention period during which to search for integration errors.

You can also specify a custom time range with which to filter the search for errors with the From and To buttons at the bottom. This functionality also exists on other monitoring pages such as the Track Instances, Track Runs, or Integrations page. See Step 3 of Track Business Identifiers in Integrations During Runtime for details.

5. View details about a specific error in either of two ways:
   a. Click View Error in the column at the right.

   A summarized version of the error message for easy understanding and resolution is displayed.

   b. If a more detailed description of the error message is required, expand Detail Error Message.

   or

   a. Click the business identifier included in the failed instance.

   The failed integration instance is displayed.
b. In the upper right corner, click View Errors.

c. If a more detailed description of the error message is required, expand Detail Error Message.

d. If you select View Audit Trail, the error in the context of the audit trail is displayed to assist in identifying where the error occurred in the overall processing.
Resubmit Failed Messages

You can manually resubmit failed messages. Resubmitting a failed message starts the integration from the beginning.

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 resubmittals 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 navigation pane, click Integrations, then click the < arrow next to Designer.
2. Click Monitoring, then click Errors.
3. From the dropdown list, select the time period or retention period during which to search for integration errors.
4. Resubmit errors in either of two ways:
   a. Click Select All to resubmit all errors or select the check boxes of individual instances to resubmit.
   b. Click Resubmit.
   or
   a. Find the specific integration to resubmit.
   b. Click at the far right.
5. Click Yes when prompted to confirm.

Discard Errors

You can discard errors based on the integration in which they occurred. A discarded error message is removed from the Errors page and can be seen in a discarded state on the Tracking page. You cannot perform any further operations on a discarded message, including recovery. After a certain time period, the error message is permanently deleted from the server.

To discard errors by integration name:
1. In the navigation pane, click Integrations, then click the < arrow next to Designer.
2. Click Monitoring, then click Errors.
3. From the dropdown list to the right, 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.

4. Discard errors in either of two ways:
a. Click **Select All** to discard all errors or select the check boxes of individual errors to discard.

b. Click **Discard**.

or

a. Find the specific error to discard.

b. Click ✖ at the far right.

5. Click **Yes** when prompted to confirm.

## 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 navigation pane, click **Integrations**, then click the < arrow next to **Designer**.
2. Click **Monitoring**, then click **Errors**.
3. Submit a failed message for recovery.
4. From the dropdown list at the top left, select **Error Recovery Jobs**.

![Error Recovery Jobs](image.png)

5. In the **Search** field, enter a partial or complete job ID of the error message recovery process.

6. View the status of message recovery:
   
   • **Received**: Recovery is still in progress.
   • **Success**: Recovery succeeded.
   • **Errors**: Recovery failed.

## View the Audit Trail of Failed Integration Instances

You can view the audit trail of a failed integration instance. This enables you to see where an integration error occurred in the message flow.

To view the audit trail of a failed integration instance:

1. In the navigation pane, click **Integrations**, then click the < arrow next to **Designer**.
2. Click **Monitoring**, then click **Errors**.
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 Audit Trail.

The audit trail 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 navigation pane, click Integrations, then click the < arrow next to Designer.
2. Click Monitoring, then click Errors.
3. To search for a specific business identifier, enter the exact business identifier value in the search field, then click Search. For example, if business identifier OrgId has a value of test2, enter test2.
   Any business identifiers with the specified value are displayed.
4. To view details about the business identifiers included in a failed integration instance, click the business identifier.

   The integration instance is displayed.

5. Click ➤ 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.

View the Message Payload of a Failed Integration Instance

You can view the message payload of a failed integration instance.

To view the message payload of a failed integration instance:

1. In the navigation pane, click Integrations, then click the < arrow next to Designer.
2. Click Monitoring, then click Errors.
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 Payload. The audit trail shows the message payload of the integration instance.

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
- Track Business Identifiers in Integrations in Which Routing Paths Are Defined
- Filter the Display of Business Identifiers in Integrations

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 Tracking page during runtime. These fields are only available for tracking on the Tracking 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 navigation pane, click **Integrations**, then click the < arrow next to **Designer**.
2. Click **Monitoring**, then click **Tracking**.
3. From the dropdown list, select the time period or retention period during which to search for business identifiers in messages.

You can also 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 Integrations, Track Runs, 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.

![Image of tracking results](image-url)
4. Use the **Search** field to search for values across both primary identifiers and secondary identifiers.

   a. To search on primary and secondary identifier values, enter a value, and click **Search**. For this example, **James** is entered. In the search results, **Instance ID 13**, **Instance ID 12**, and **Instance ID 11** that are returned have a primary value of **James**. **Instance ID 14** with a secondary identifier value of **Mark.James@asc.com** is also returned.

   ![Search results](image1)

   b. To search only on primary business identifier values, enter the value as `primary: value` or `Primary: value` in the **Search** field.

   ![Search input](image2)

5. 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 audit trail, 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.
6. From the menu, select **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. For orchestrated integrations, the message flow through each activity (for example, any defined switch activity branches) is shown.

7. 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.

**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 Tracking page is case sensitive.

**Track Business Identifiers in Integrations in Which Routing Paths Are Defined**

If the integration in which you defined business identifiers also includes definitions for routing paths, you can view the value of the business identifier, the status of the routing path taken based on the business identifier value, the routing expression logic in the blue header above the integration, and the status of the overall integration flow. The status of the routing path taken and the overall integration flow are indicated by color (green indicates success and red indicates failure).

To track business identifiers in integrations in which routing paths are defined:

1. In the navigation pane, click **Integrations**, then click the < arrow next to **Designer**.
2. Click **Monitoring**, then click **Tracking**.

3. From the menu, select the time period during which to search for business identifiers in messages.

4. Click the business identifier to access a graphical display of the integration instance.

The page shows (from top to bottom) the value of the business identifier, the status of the routing path taken based on the business identifier value, the routing expression logic in the blue header above the integration, and the status of the overall integration flow.

## Filter the Display of Business Identifiers in Integrations

You can filter the display of business identifiers on the Tracking page during runtime.

To filter the display of business identifiers in integrations:

1. In the navigation pane, click **Integrations**, then click the < arrow next to **Designer**.

2. Click **Monitoring**, then click **Tracking**.

3. Click ![filter](image) and select a display method. The primary identifier is not displayed as a filter. All other identifiers are displayed as filters.

   - **Message > Additional Business Identifiers**
     a. Begin entering the integration name to display names that begin with those letters or select an integration from the dropdown list to see the available tracking fields.

     Any business identifiers associated with the integration are displayed.

     b. To filter, enter the exact business value for the business identifier.

     c. Click **Set Filter**.

   - **Integration**
     a. Begin entering the integration name to display names that begin with those letters or select an integration from the dropdown list to see the available tracking fields.

     b. Click **Set Filter**.

   - **Run Id**
     a. Enter the run ID value for the scheduled integration.

     b. Click **Set Filter**.

   - **Instance Id**
     a. Enter the instance ID value for the scheduled integration.

     b. Click **Set Filter**.

   - **Message State**
     a. Select to show completed, failed, or aborted integrations.
View Preinstalled Adapters

You can view the adapters preinstalled with Oracle Integration.

1. In the navigation pane, click Integrations, then click Adapters.

   The Adapters page show the following categories of adapters in Oracle Integration
   • Preinstalled: Adapters included with your instance of Oracle Integration (for example, adapters to which you have subscribed and adapters that are automatically included).
   • Marketplace: Adapters included with the packages imported from the Oracle Marketplace.

2. Click the adapter name or select View from the menu at the far right to display information about the adapter.

3. If you want to delete an adapter, select Delete from the menu at the far right. You cannot delete adapters identified as Preinstalled.

Send Service Failure Alerts, System Status Reports, and Integration Error Reports by Notification Emails

You can notify users by email with hourly or daily reports about total messages received, total messages processed, successful messages, failed messages, and successful message rate; with service failure alerts (for example, when a runtime or storage service is down or an agent becomes unavailable); or with a detailed report every five minutes when an integration failure occurs.

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.

To send service failure alerts, system status reports, and integration error reports by email:

1. In the navigation pane, click Integrations, then click the < sign next to Designer.

2. Click Settings, then click Notifications.

3. Select when to send an email notification:
   • Send an alert when a service failure 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
- Send a detailed report every five minutes (only if an error occurred)

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 audit trail details, integration name, version number, and other information

The email includes a link to the integration instance diagram that appears under the Tracking 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.

4. In the Distribution list field, enter the email address to which to send the reports and alerts. You can specify multiple email addresses, each separated by commas.

5. For the Sender Email Address field, click + in the lower right corner to add approved senders. 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. However, only a single email address can be selected to send system notifications.

6. Click Save.

7. 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 revert to your previously saved configuration, click Revert.
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.

**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 navigation pane, click **Integrations**, then click the `<` sign next to **Designer**.
2. Click **Settings**, then click **Logging Levels**.
   
   The loggers available for configuration are displayed. If the logger level is inherited from a parent logger, that information is displayed.
3. Change the logger level of any logger.
4. To revert to the previous setting before clicking **Save**, click **Revert**.
5. 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.
6. Click the **Home** icon, then click the **Monitoring** diagram.

   **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**.

7. Click **Download Diagnostic Logs**.
   
   Note that the downloaded ZIP file name includes the WebLogic Server domain name (the same as your POD name) prepended to the front. For this example, the POD name of **domain_ics_expanded** is prepended to the front: `domain_ics_expanded-diagnostic-logs-number.zip`. This enables you to know the POD from which the file is being downloaded.

   You can also get the POD name from inside the **AdminServer-diagnostic.log** file that is included in the ZIP file.
8. Unzip the file, and within the expanded directory structure, open AdminServer-diagnostic.log.

9. 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 (for this example, domain_ics_expanded is also the name).

   [2016-11-09T12:00:00.413-08:00] [AdminServer] [NOTIFICATION]
   [ICS-10071] [oracle.ics] [tid: [ACTIVE].ExecuteThread: '5' for queue:
   'weblogic.kernel.Default (self-tuning)'] [userId: <anonymous>] [ecid:
   f1xa8364-21dc-4c08-8843-a57b23b9bde2-000051f5,0] [APP: icswebapp]
   Weblogic domain name is "domain_ics_expanded".

   An environment.txt file is part of the downloaded ZIP file. It 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).

1. In the navigation pane, click 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.

   Incident report successfully created with ID: "number". You can
download it right now or later from the Monitoring dashboard.

5. Click the  icon, then click Integrations.

6. In the navigation pane, click the < arrow next to Designer.

7. Click Monitoring, then click Dashboards.

8. Click Download Incident.

9. Enter the incident ID number, then click Download.

10. 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.

1. In the navigation pane, click Integrations, then click the < sign next to Designer.

2. Click Settings > Database.

   The Database Configuration page is displayed.

3. View the current database space usage, the quiesced and unquiesced mode thresholds, and your proximity to the database notification warning threshold (the threshold at which you can receive a notification alert email).

4. Update threshold values as appropriate, then click Save to activate your updates.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quiesced Mode Threshold</td>
<td>Click the appropriate up or down arrow to modify the threshold mode value. Runtime operations are shut down and the database enters quiesced mode when this space usage threshold value is reached.</td>
</tr>
<tr>
<td>Un-Quiesce Threshold</td>
<td>Click the appropriate up or down arrow to modify the threshold mode value. Runtime operations resume automatically and the database enters unquiesced mode when this space usage threshold value is reached.</td>
</tr>
</tbody>
</table>
Field | Description
--- | ---
**Warning Threshold** | Click the appropriate up or down arrow to modify the warning threshold. You are notified when this space usage threshold is reached. The warning threshold value must be less than the quiesced mode threshold value and greater than the unquiesced threshold value. See *Send Service Failure Alerts, System Status Reports, and Integration Error Reports by Notification Emails*. Automatic purging also begins when this threshold is reached. You configure automatic purging on the **Auto Purge** tab of this page.

5. **Click the Nightly Purge tab.**

You can purge instance data of successful and faulted integration instances that have been inactive over a specified period of time and reclaim the database space. Every midnight, the runtime data older than this retention period is purged. These settings take effect when you click **Save**.

a. Specify the retention period for purging successful and faulted instances from the system. You can choose to keep the faulted instances data around longer to attempt fault recoveries. Every midnight, runtime data older than the retention periods is purged. This setting goes into effect when you click **Save**. The **Time Unit** list enables you to select **Days** (default is three) or **Hours** (default is three) for the purge period (in case you need to revert to the defaults by clicking **Revert**).

![Diagram](image)

b. **Click Purge Now** in the top right corner of the page to invoke a dialog to manually perform an ad-hoc purge of the database. This button is available on all three tabs. The next nightly purge still runs as scheduled.
c. Specify how long to keep the successful and faulted instance data. For example, if your usage is very high, you may choose to purge all successful instances within one hour of completion. Instance data older than the retention values specified is purged immediately when you click Purge Now.

6. Click the Auto Purge tab.
a. Click Auto Purge Enabled to automatically purge the database without reclaiming database space when the Warning Threshold value you specified on the Database Space tab is reached. An automatic purge and a reclaim of database space also occurs when the Quiesced Mode Threshold value is reached.

b. Specify the retention period values for purging successful and faulted instances from the system or click Match nightly purge settings to use the same purge settings as are used for the nightly purge.

c. Click Save. When the Warning Threshold value you specified on the Database Space tab is reached, an automatic purge occurs.
When you add an adapter to a trigger (source) or an invoke (target) in an integration, a wizard prompts you to configure how the data is processed for that connection, including the type of operation to perform, the business objects or fields to use, and so on. The properties you can configure vary by each type of adapter. Click any of the topics below to read more about the properties you can configure for each type of adapter.

Topics:

• Configure Trigger Oracle Integration Messaging

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.

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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.
Troubleshoot Oracle Integration

Topic:
- Conflict with Using Variables of String Data Type and Applying Mathematical Comparator Expressions
- Reduce Network Calls that Slow Down Processes Caused By Writing Record By Record
- 403 Error when Accessing an API Protected using OAuth
- Troubleshoot Integration Activations
- Invoke Subprocess Integrations Using a Single REST API Connection
- Troubleshoot Certificate Import Failures
- Trigger Connection Cannot Receive an Empty Array in JSON if the Corresponding JSON Received Has an Empty JSON

Note:
Adapter troubleshooting information is provided in the individual adapter guides. See the Adapters Home Page in the Oracle Help Center.

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.

### 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 Integration Activations

If your integration fails to activate, review the following failure issues and corrective actions.

<table>
<thead>
<tr>
<th>Activation Failure Causes</th>
<th>Corrective Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>WSDL issues:</td>
<td></td>
</tr>
<tr>
<td>• Duplicate schema issues in the WSDL</td>
<td>1. Check the WSDL for duplicate schemas.</td>
</tr>
<tr>
<td>• WSDL/schema validation failures</td>
<td>2. Check that the WSDL is valid.</td>
</tr>
<tr>
<td>• WSDL validation failures</td>
<td>3. If necessary, correct the WSDL in the application.</td>
</tr>
<tr>
<td>• Deployment time out issues.</td>
<td>4. Reconfigure the integration by regenerating the endpoints.</td>
</tr>
<tr>
<td>• Retry the activation.</td>
<td>5. Retry the activation because the server may be slow to react.</td>
</tr>
<tr>
<td>Activation Failure Causes</td>
<td>Corrective Actions</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>----------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Database issues:</strong></td>
<td>1. Check that the database is running and has enough space.</td>
</tr>
<tr>
<td>• Reached the size limit:</td>
<td>2. If not, the database administrator must increase the database size limit or</td>
</tr>
<tr>
<td></td>
<td>connection pool size.</td>
</tr>
<tr>
<td></td>
<td>3. Retry the activation.</td>
</tr>
<tr>
<td>• Database is down.</td>
<td></td>
</tr>
<tr>
<td>• Connection pool issues:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Could not create pool connection for</td>
<td></td>
</tr>
<tr>
<td>datasource 'SOALocalTxDataSource'. The</td>
<td></td>
</tr>
<tr>
<td>DBMS driver exception was: Listener</td>
<td></td>
</tr>
<tr>
<td>refused the connection with the following</td>
<td></td>
</tr>
<tr>
<td>error: ORA-12516, TNS:listener could</td>
<td></td>
</tr>
<tr>
<td>not find available handler with matching</td>
<td></td>
</tr>
<tr>
<td>protocol stack</td>
<td></td>
</tr>
<tr>
<td>A connection is down or was modified.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Managed server unavailable.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1. The administrator must restart the server.</td>
</tr>
<tr>
<td></td>
<td>2. Retry the activation.</td>
</tr>
</tbody>
</table>
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/338c94f1-d3ee-45b9-bbf8-43c9203df851/SCA-INF/bpel/StageCurrentGTHROFileForEachBpel/src/orabpel/stagecurrentgthrofileforeachbpel/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.

```xml
<features>
  <feature enabled="true">sdk.wsdl_regeneration</feature>
</features>
```

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.

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
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:

```xml
<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:

```xml
<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.