

Orchestration Integration Actions

Objectives

After completing this lesson, you should be able to:

- Leverage the design canvas view options
- Use actions in orchestration style integrations
- Be familiar with Stage File action operations
- Invoke OIC resources including local integrations and processes
- Design with looping constructs and conditional branches

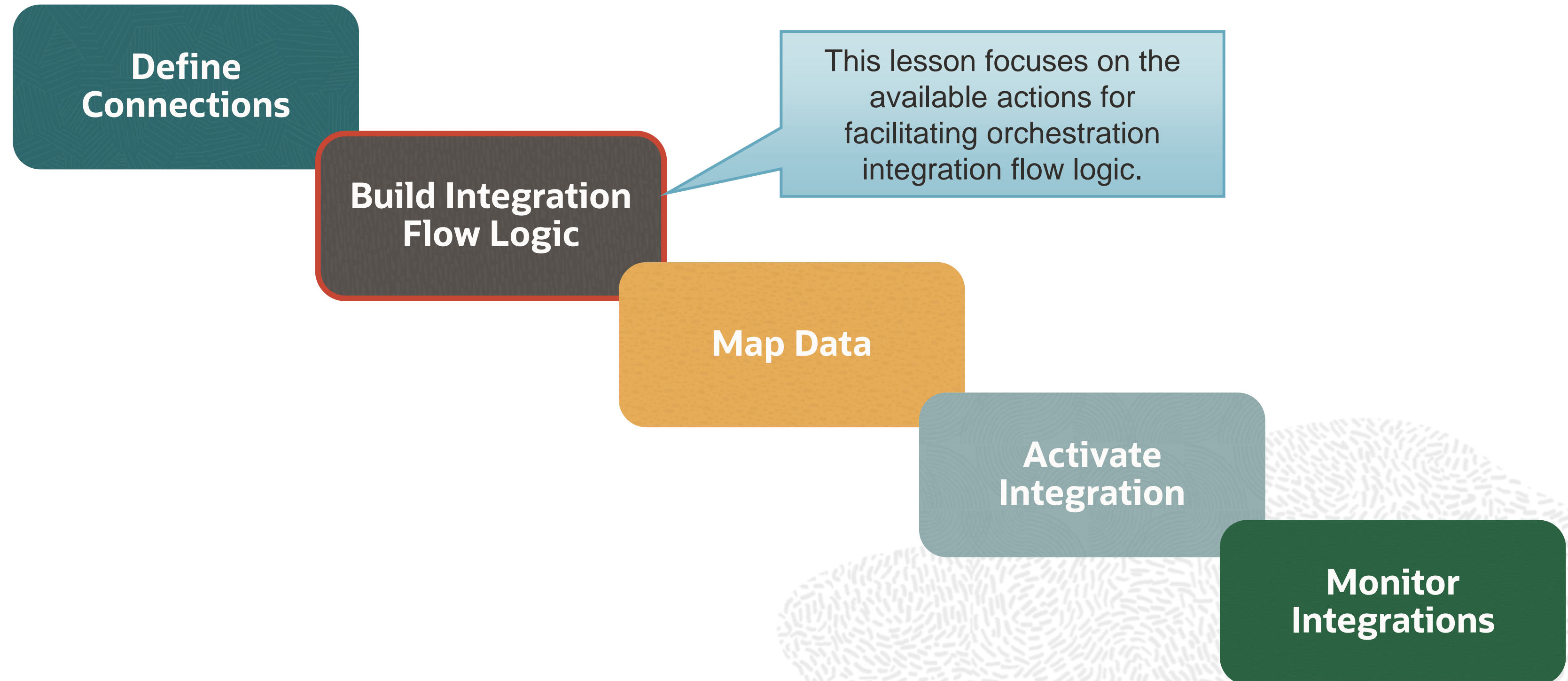


Agenda

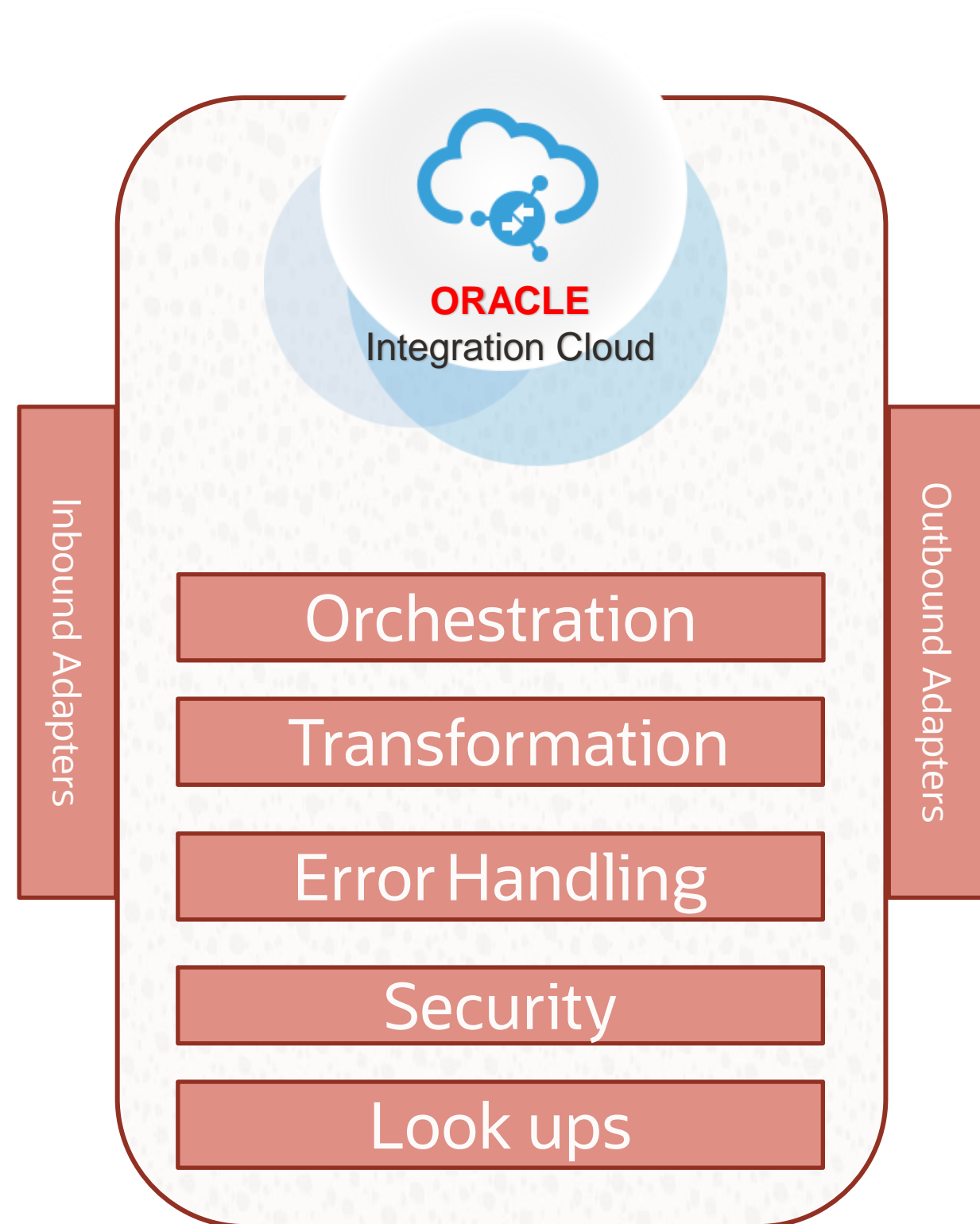
- Orchestration Design Canvas Editing Options
- Orchestration Menu Actions
 - Data Section
 - Call Section
 - Collection Section
 - General Section
 - End Section



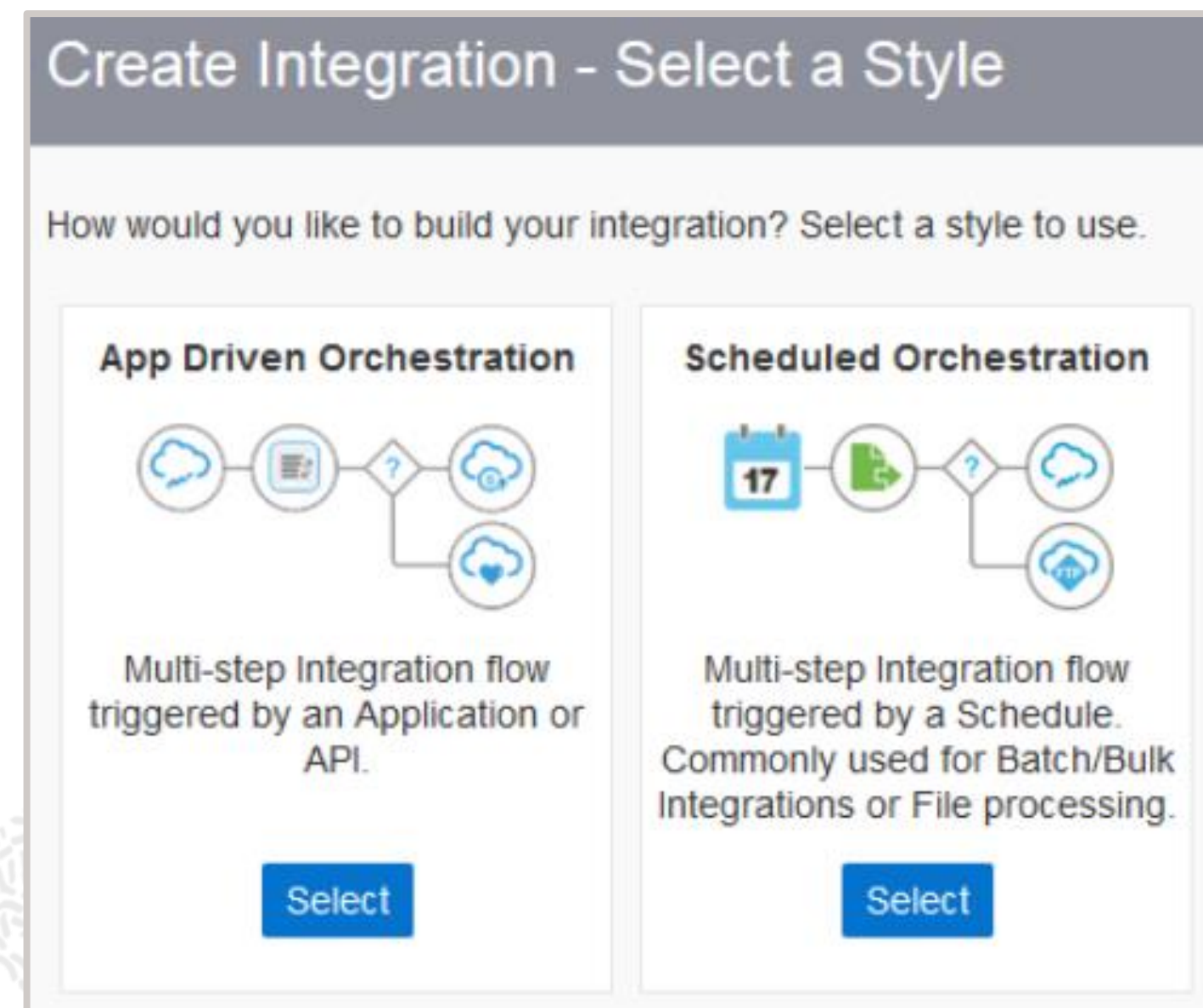
Integration Development (Review)



Orchestration Styles (Review)



- **App Driven** (Interface invoked by a client application)
- **Scheduled** (Executes automatically or on-demand)



Editing View Options

- There are two views to choose from:

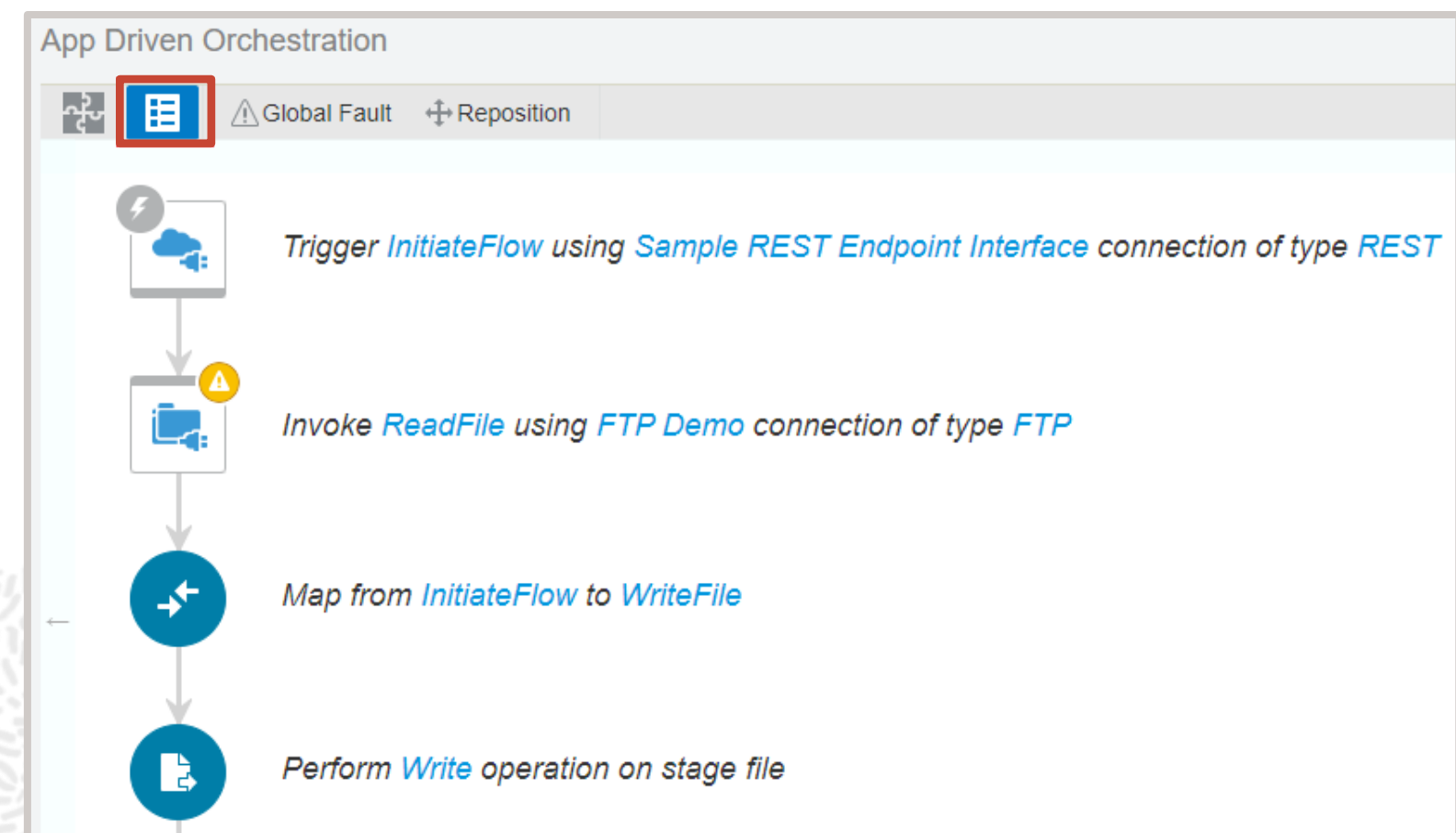
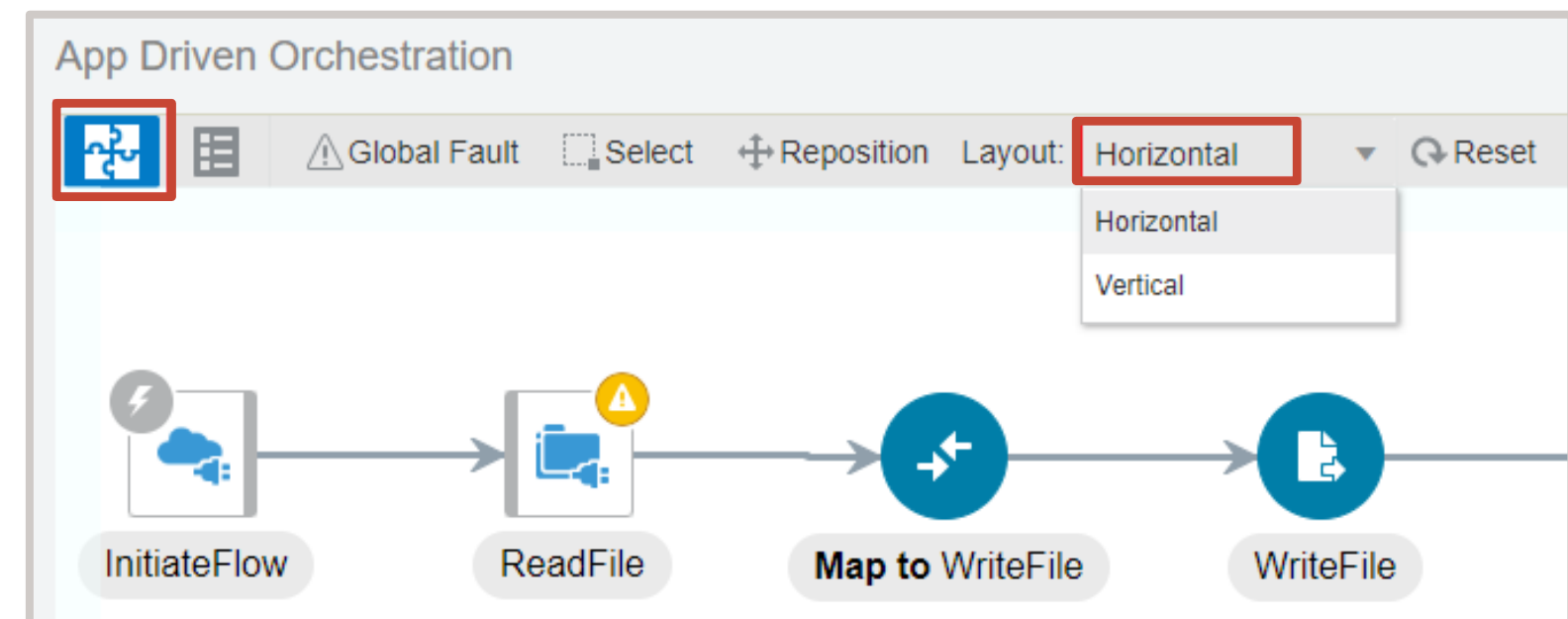


- Canvas View
 - Vertical or Horizontal



- Pseudo View
 - Vertical only, with indented child nodes

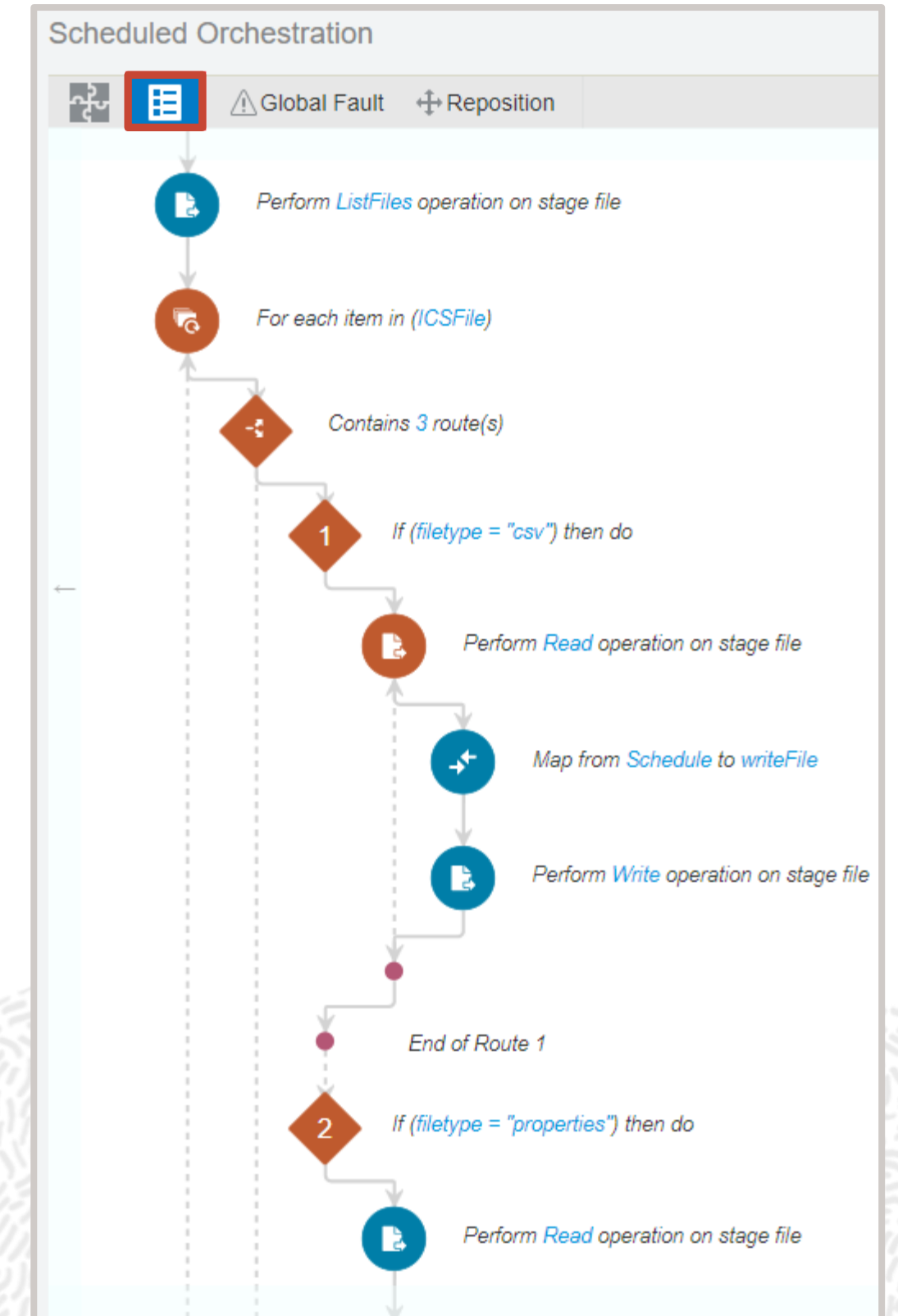
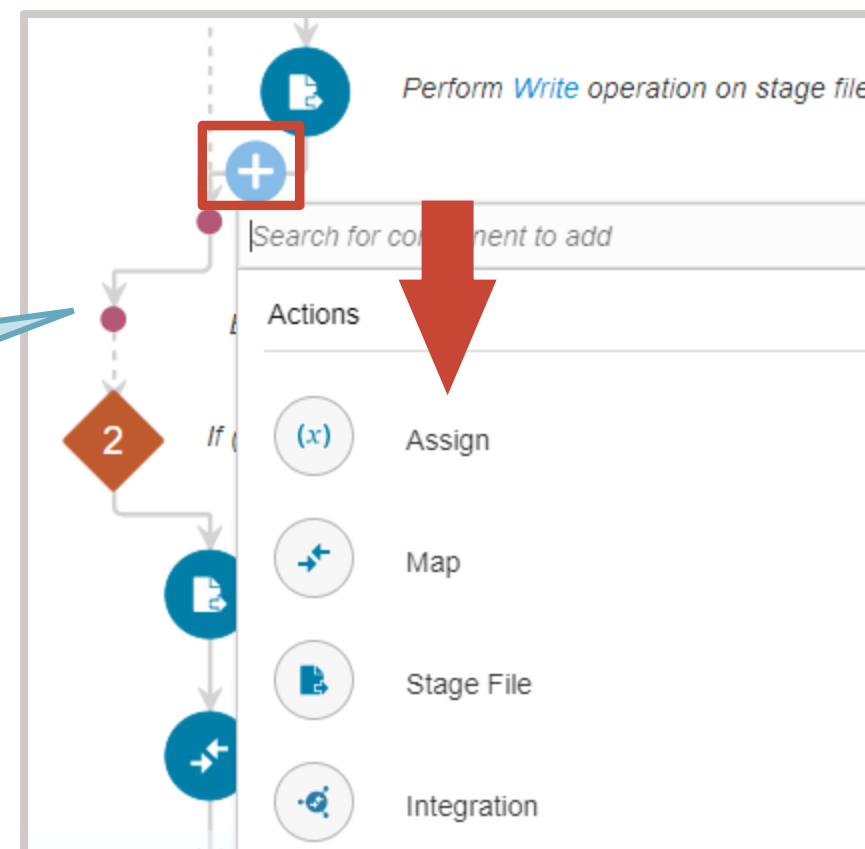
- You can navigate between these views at any time.



Pseudo View

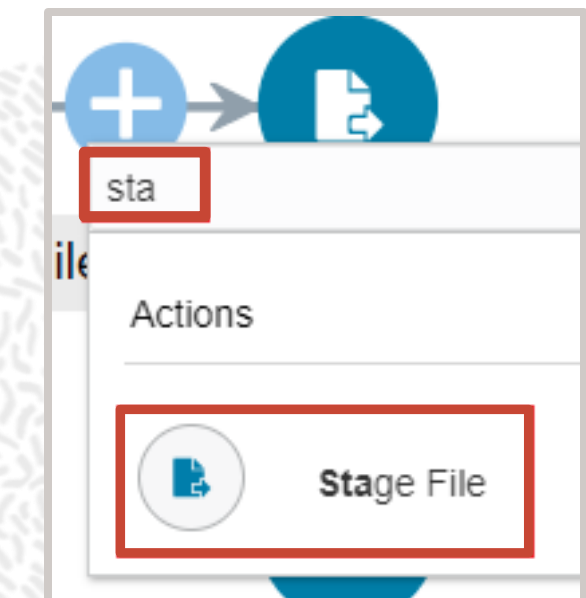
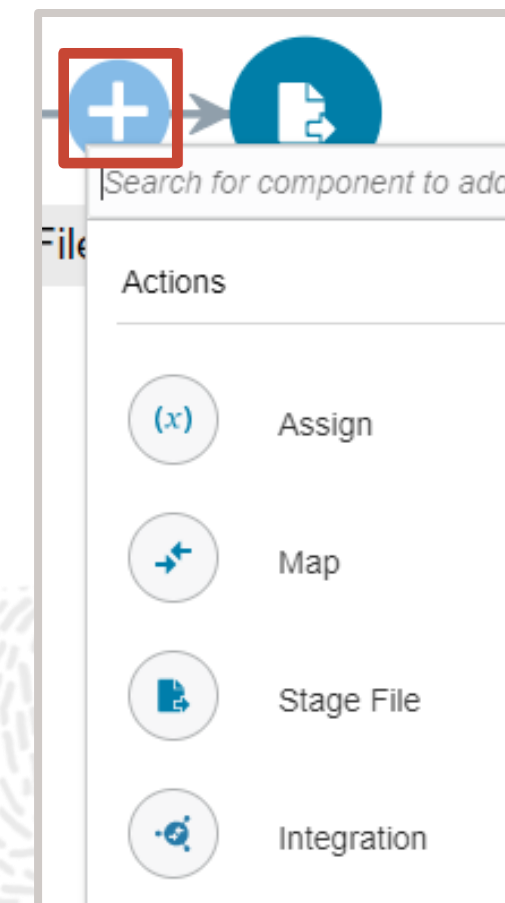
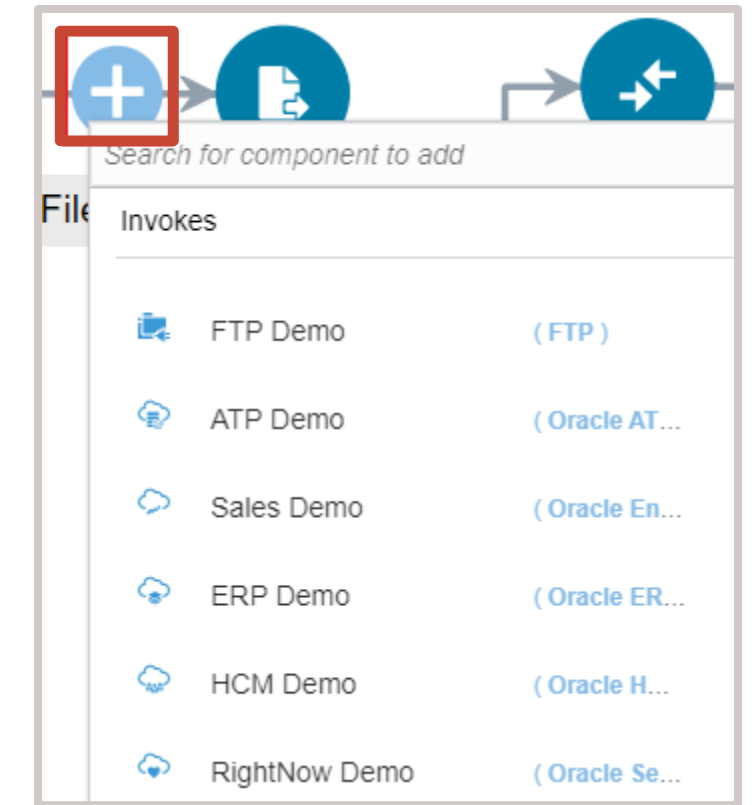
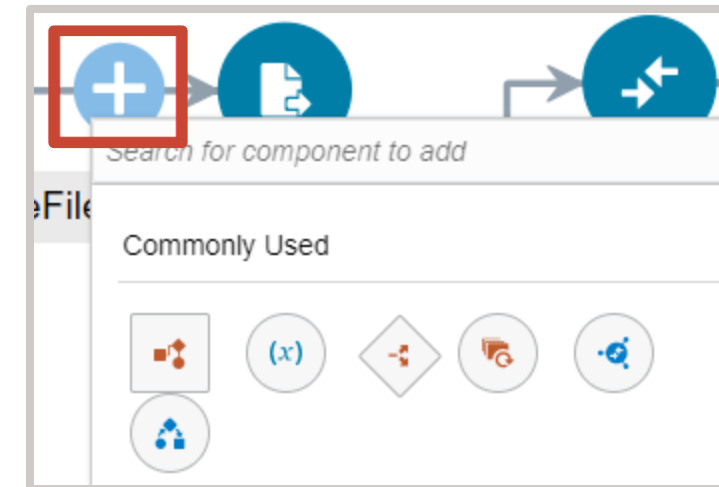
- In this view, the integration is shown vertically with indented child nodes.
- For each node, it shows the details.
 - This helps you to easily understand the integration without having to open and drill down to each node.
- You can use the inline menu to add new invokes or actions.

The inline menu can also be used in the Canvas view.



Using the Inline Menu

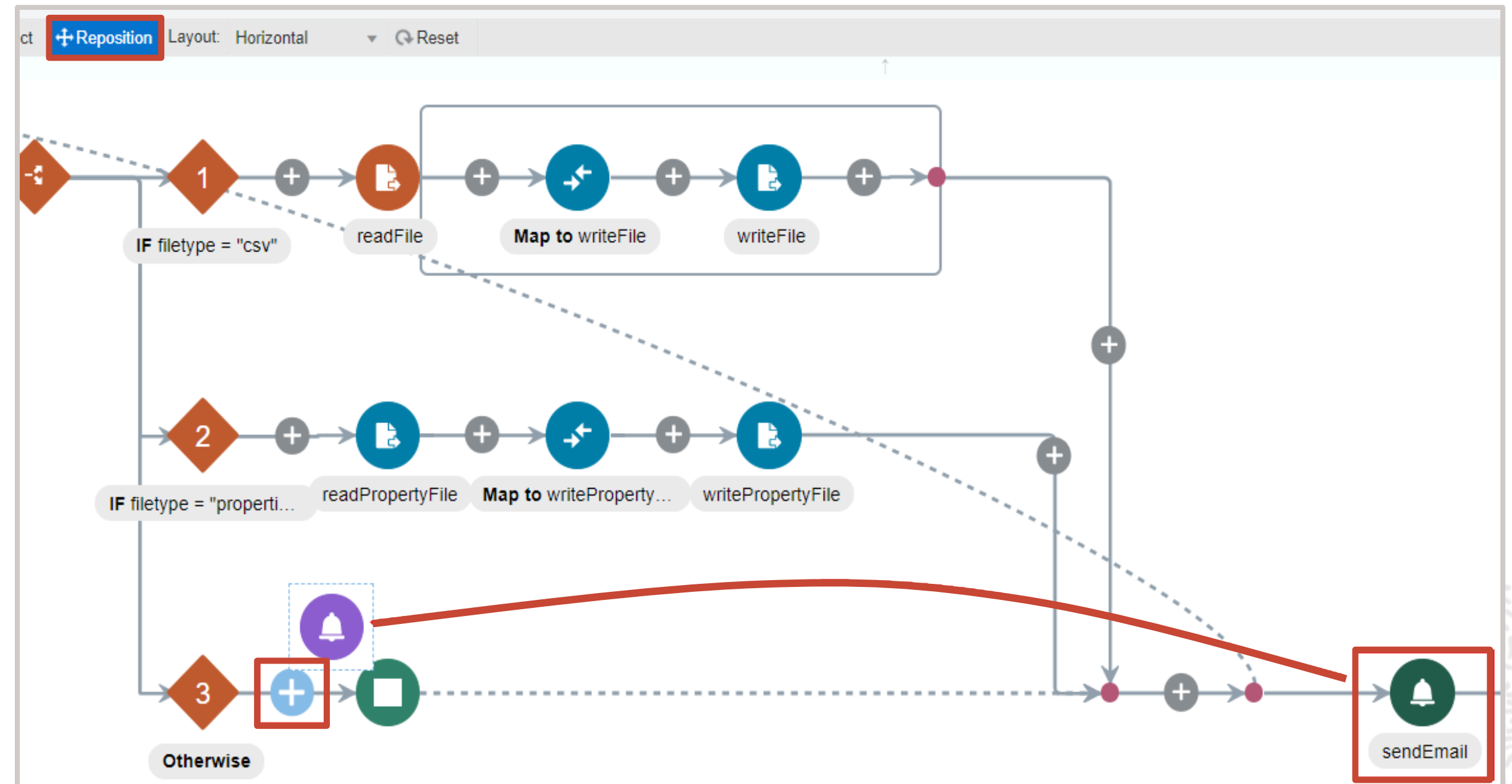
- Hover over a location, and then click the blue + icon. The menu appears.
 - Only actions that are allowed to be placed at that location will appear.
- The menu has three sections:
 - Commonly Used components
 - Invokes
 - Actions
- The menu also provides the ability to search for a particular Invoke or Action.
- Partial matches will be returned as well.



Repositioning Invoke and Action Nodes

If you need to move a node to a different location within the integration flow:

- Click the **Reposition** button.
- Drag the node to any location indicated by a plus (+) icon.
- After a node is moved, click the **Reposition** button again to return to normal editing mode.



Agenda

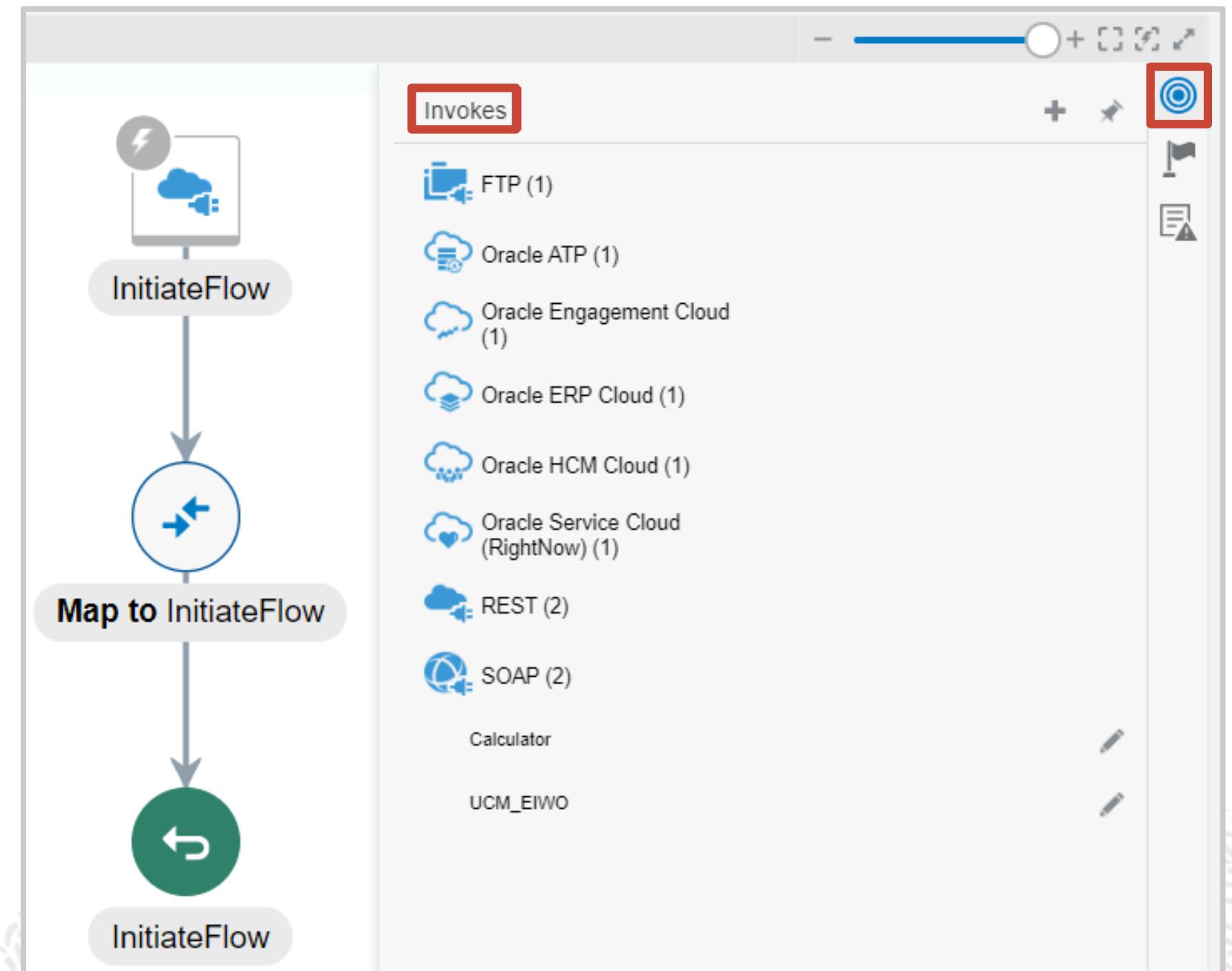
- Orchestration Design Canvas Editing Options
- Orchestration Menu Actions
 - Data Section
 - Call Section
 - Collection Section
 - General Section
 - End Section



Using Outbound Invokes (Review)

After the initial inbound trigger has been configured, the Triggers section is replaced by the Invokes section.

- Multiple outbound connections can be used to invoke external resources as required by the business use case.
- Invokes can be placed within:
 - The main flow path
 - The global fault handler
 - For-Each loops
 - While loops
 - Branches of a Switch activity
 - Scopes
 - Scope fault handlers



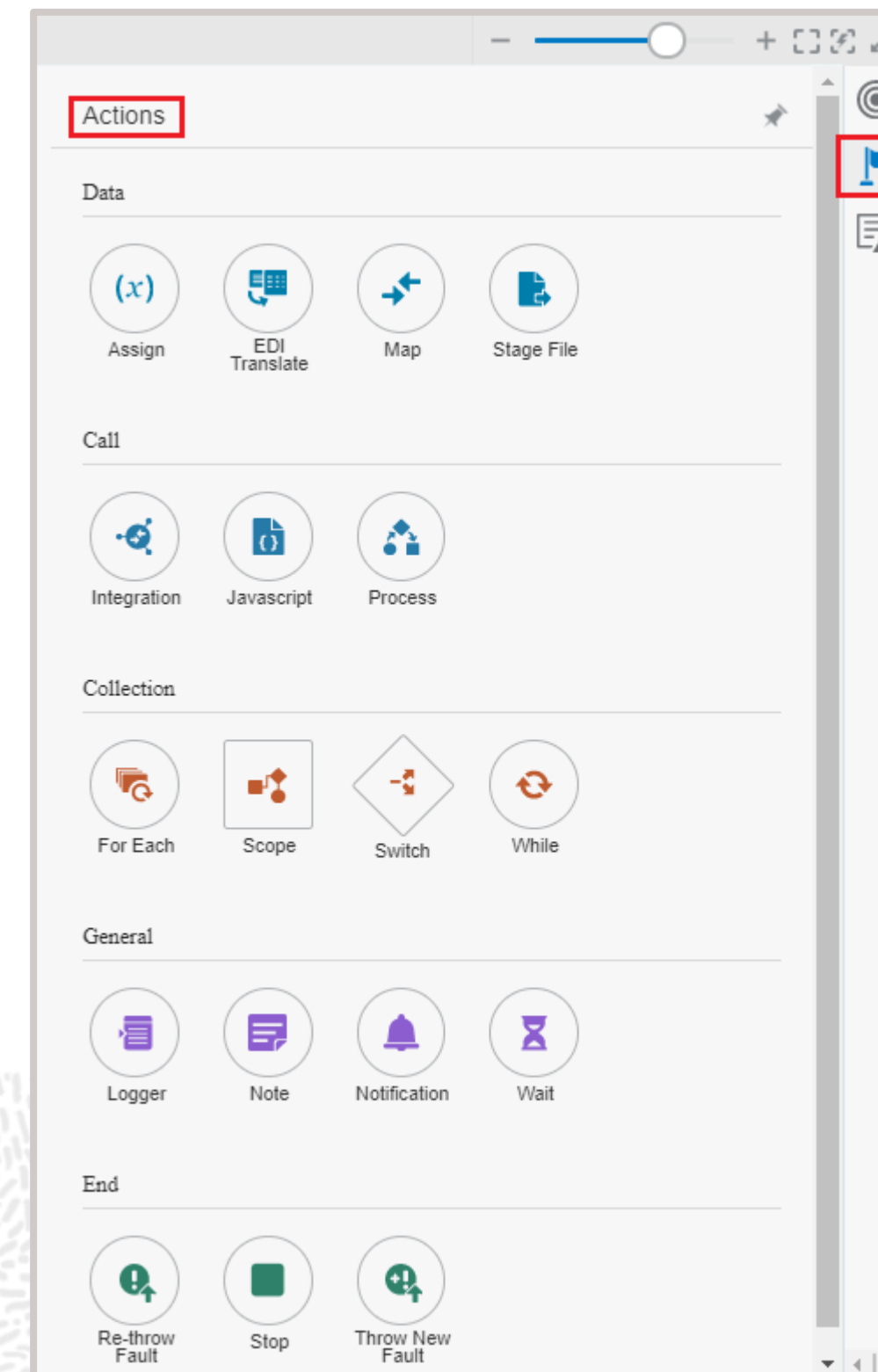
Actions Menu

Actions are organized into five sections:

- Data
- Call
- Collection
- General
- End

Actions can be placed within:

- The main flow path
- The Global fault handler
- For-Each loops
- While loops
- Branches of a Switch action
- Scopes
- Scope fault handlers



Agenda

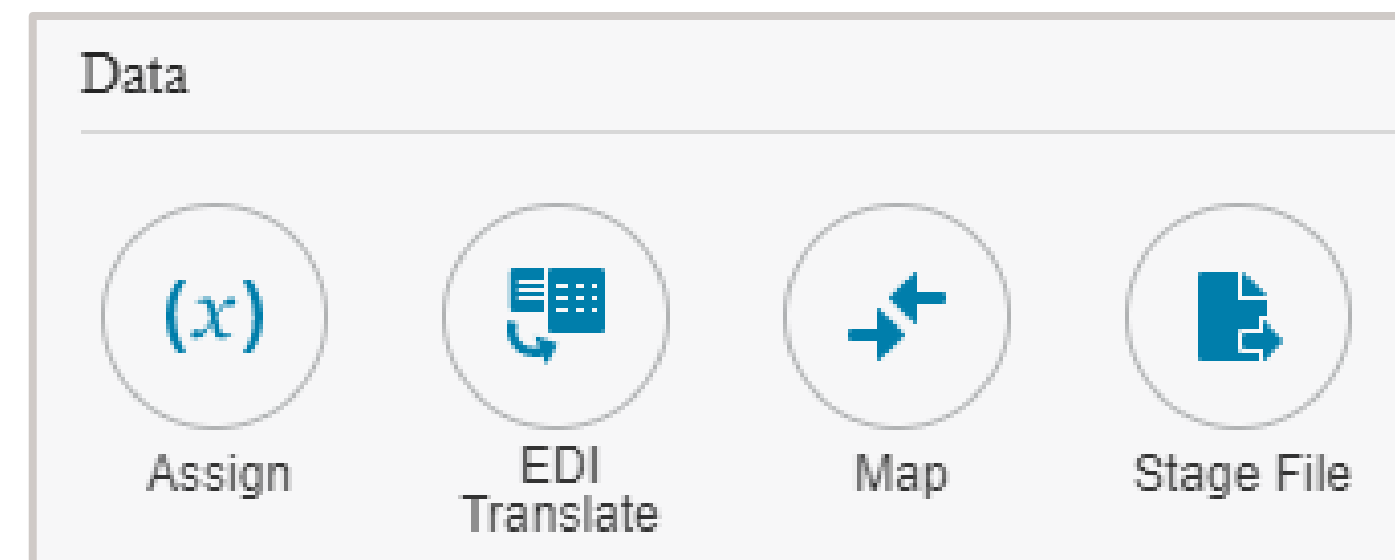
- Orchestration Design Canvas Editing Options
- Orchestration Menu Actions
 - Data Section
 - Call Section
 - Collection Section
 - General Section
 - End Section



Data Section

Selecting the **Actions** menu provides the following functionality in the **Data** section:

- **Assign:** Assign values to one or more scalar variables using the expression editor.
- **EDI Translate:** Translate an XML message to or from the Electronic Data Interchange format.
- **Map:** Add an Ad-Hoc mapper to the Integration.
- **Stage File:** It launches a wizard for configuring file operations to include List, Read, Write, Zip, and Unzip.



Assign Action...



Use the **Assign** action to create variables or to assign new values to existing variables.

- Provide a variable name, data type, description and value.

The expression builder is used to assign values.

1 Click the **+** icon to add a component.

2 In the **Create Action** dialog, enter a unique name and description for the action, then click **Create**.

3 In the **InitializeVars** configuration window, click the **+** icon to add a variable.

4 In the variable configuration table, set the variable name and data type.

5 Click the edit icon in the **Value** column to enter an expression.

Variable	Data Type	Description	Operation	Value
(x) email_subject	string	To be used in the Subject field		Add an expression

Expression

```
concat("Action regarding ", /nsmpr10:process/nsmpr10:Account/nsmpr9:OrganizationName, " organization")
```

Expression Summary

```
concat( "Action regarding ", OrganizationName, " organization")
```

Assign Action (continued)



Variable	Data Type	Description	Operation	Value
(x) email_subject	string	To be used in the Subject field		concat("Action regarding ", OrganizationName, " organization")
(x) attempts	string	Processing attempts		"1"

Click “+” to create additional variables.

Close

Validate

7

Click **Validate** then **Close** to complete the action.

(x)

InitializeVars

- Use additional **Assign** actions to update values of existing variables.

UpdateAttempts ora028 RN Incidents For OSC (1.1.0)

(x) Assign

Assign variables to your integration. You can assign values to variables using the the assignments as needed. These values can be accessed later on from other c

Add at least one named variable and specify its value by adding an expression.

Variable

Data Type

(x) attempts

string

attempts

email_subject

tracking_var_2

tracking_var_3

Variable	Data Type	Description	Operation	Value
(x) attempts	string	Update number of processing attempts		string(integer(\$attempts) + 1.0)

(x)

UpdateAttempts

Note: Variables created inside a **Scope**, a **For-Each** loop, or a **While** loop will not be visible outside that scope or loop.



EDI Translate Action



B2B for OIC works in an orchestrated integration through use of the EDI translate action.

- When you add this action to an integration, the EDI Translate Action Wizard is invoked to guide you through configuration with the EDI X12 document standard.
- You can also select to include standard or customized document definitions in your configuration.

Create New B2B Schema

What is it called?
The Name can be changed later. The Identifier can be set only now and it must be unique.

* Name

New Business Schema

* Identifier

NEW_BUSINESS_SCHEMA

What does it do?
Description *What does it do?*

1024 characters left

* Document Standard

X12

* Document Version

4010

* Document Type

850 (Purchase Order)

Cancel

Create

Basic Info

Select Data Formats

Translate Sample Data

Summary

Select the direction in which to translate the message:

☒ Inbound EDI message to Oracle Integration Cloud message

☐ Oracle Integration Cloud message to outbound EDI message

Choose the type of EDI message

* Document Standard

EDI X12

* Document Version

4010

* Document Type

855 (Purchase Order Acknowledgment)

* Document Definition

Standard

* EDI Character Encoding

UTF8

Perform validations on input data? ☒ Yes ☐ No

Create New B2B Document

What is it called?
The Name can be changed later. The Identifier can be set only now and it must be unique.

* Name

My_PO

* Identifier

MY_PO

What does it do?
Description *What does it do?*

1024 characters left

* Document Standard

X12

* Document Version

4010

* Document Type

850 (Purchase Order)

Cancel

Create

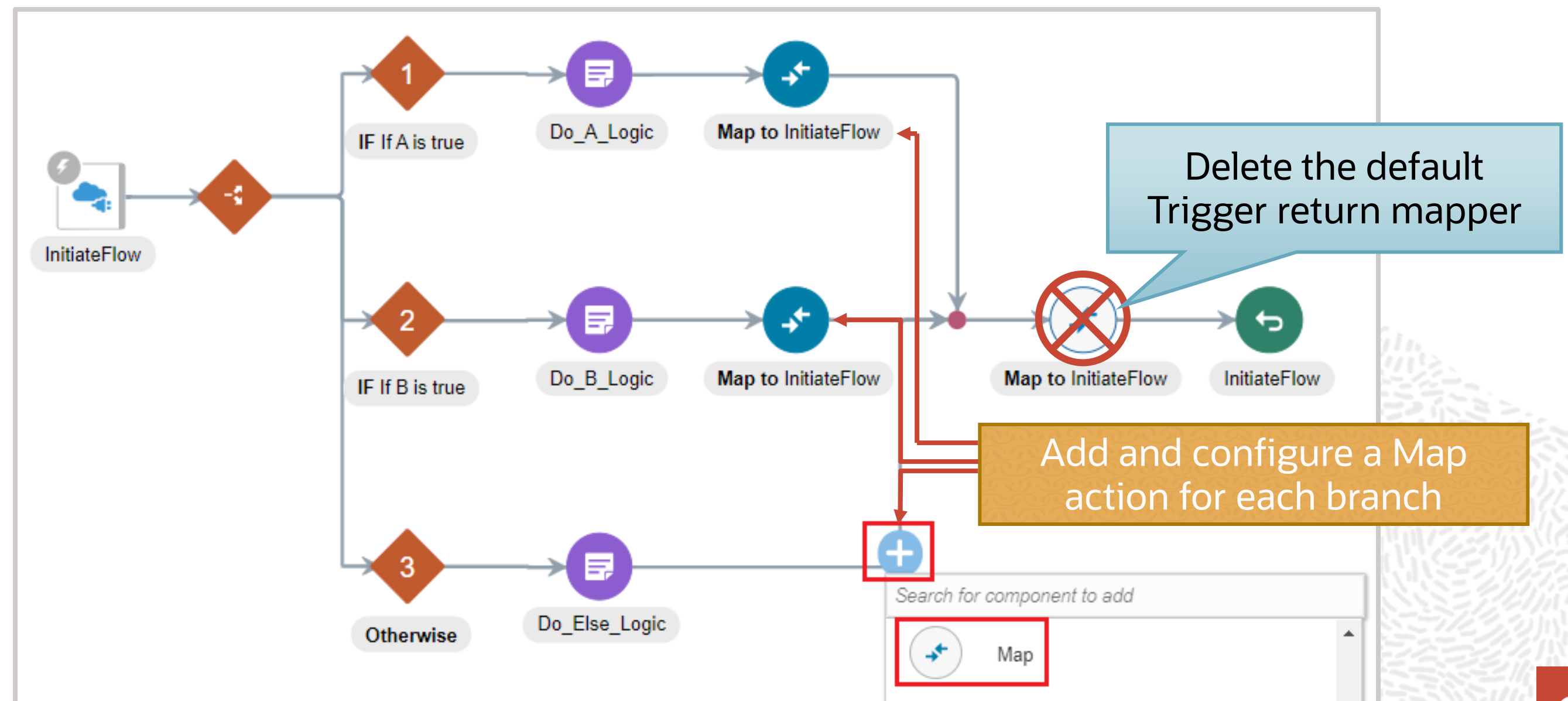


Map Action

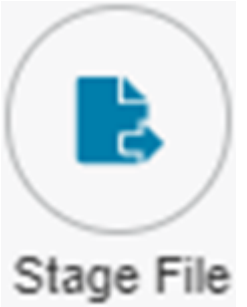


A **Map** action is automatically added after the Trigger has been configured.

- It is also added automatically after a new **Invoke** or **Call** action has been configured.
- The most common use case for needing to add an ad hoc mapper is within branches of a **Switch** activity.



Stage File Action



Stage File operations enable processing files or file references within the integration flow. Files and references are made available within the integration by using:

- An FTP Adapter's **Read** or **Download** operation or a File Adapter's **Read** operation.
- Multipart or MTOM attachments provided by SOAP and REST APIs

Start by selecting from one of 8 operations on the **Configure Operation** page:

- **Read Entire File**
- **Read File in Segments**
- **Write File**
- **Zip Files**
- **Unzip File**
- **Encrypt File**
- **Decrypt File**
- **List Files**

Subsequent configuration wizard options will vary based on chosen operation

Stage File: Read Entire File



Returns the file into memory from a file reference or from a specified directory in the OLC virtual file system (VFS)

File sizes up to 10 MB

Choose Stage File Operation

Read Entire File

Configure File Reference

Specify the File Name

\$ReadFile/nsmpr4:SyncReadFileResponse/nsmpr9:FileReadResponse/nsmpr9:FTPResponseHeader/nsmpr9:fileName

Specify the Directory to read from

\$ReadFile/nsmpr4:SyncReadFileResponse/nsmpr9:FileReadResponse/nsmpr9:FTPResponseHeader/nsmpr9:directory

Remove Trailer

None

None

Last Row

Last 'n' Rows

Use the Expression Builder to specify the file name and directory...

Choose Stage File Operation

Read Entire File

Configure File Reference

Specify the File Reference

\$GetFile/nsmpr10:ListFilesResponse/nsmpr1:ListFilesResponse/nsmpr1:FileList/nsmpr4:ICSFile/nsmpr4:FileReference

Remove Trailer

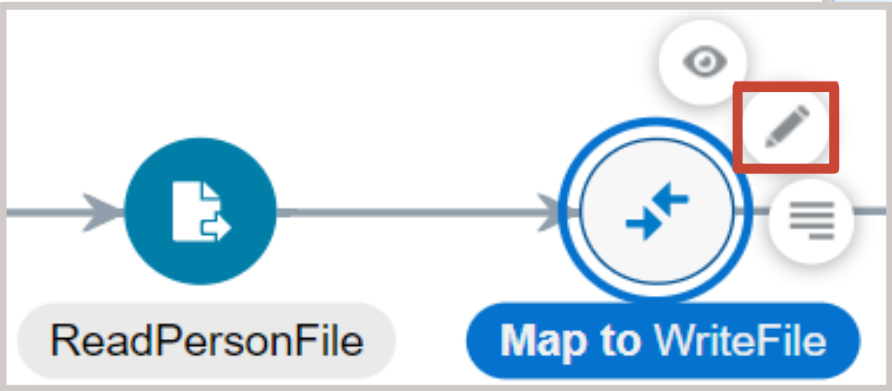
None

None

Last Row

Last 'n' Rows

...or the file reference.



The file is returned with its schema structure (available in subsequent mappers)

Source	Find...
▶ <> *execute	
▶ <> \$ReadFile	
▶ <> \$ReadFile_REQUEST	
▲ <> \$ReadPersonFile	
▲ <> *ReadResponse	
▲ <> *allpersons	
▲ <> *person	
▲ <> *personal	
<> *firstName	
<> *lastName	
<> *dob	
<> *nationalIdentifier	
▲ <> *work	
<> *employeeExternalNumber	
<> *legalEmployerId	
<> *legalEmployerName	
<> *businessUnitId	
<> *hireDate	
▲ <> *contactInfo	
<> *homePhone	
<> *email	



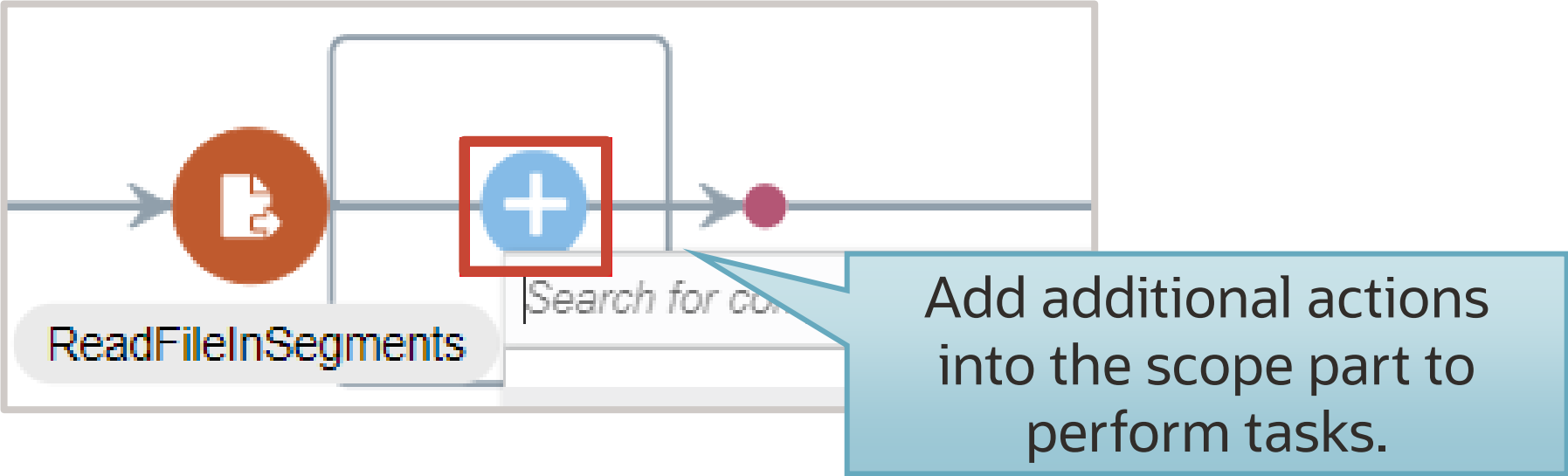


Stage File: Read File in Segments

Creates a scope part that returns the file in logical chunks (segments) from a file reference or a filename from a specified directory

- Segment size defaults to 200 records.

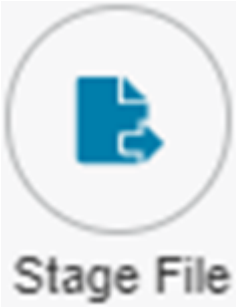
For larger files up to 1 GB



Segments can be processed sequentially or in parallel threads.

- For example:
- Add a for-each loop to process one record at a time.
 - Add “child” Stage File operations to further process each of the file chunks.





Stage File: Write File

Used to create a new file to be staged into a specified directory in the OLC virtual file system (VFS), or used to append to an existing file in the VFS.

- It is available for further processing within the scope in which it was created.
- File is automatically deleted once the integration flow instance moves outside of the scope.

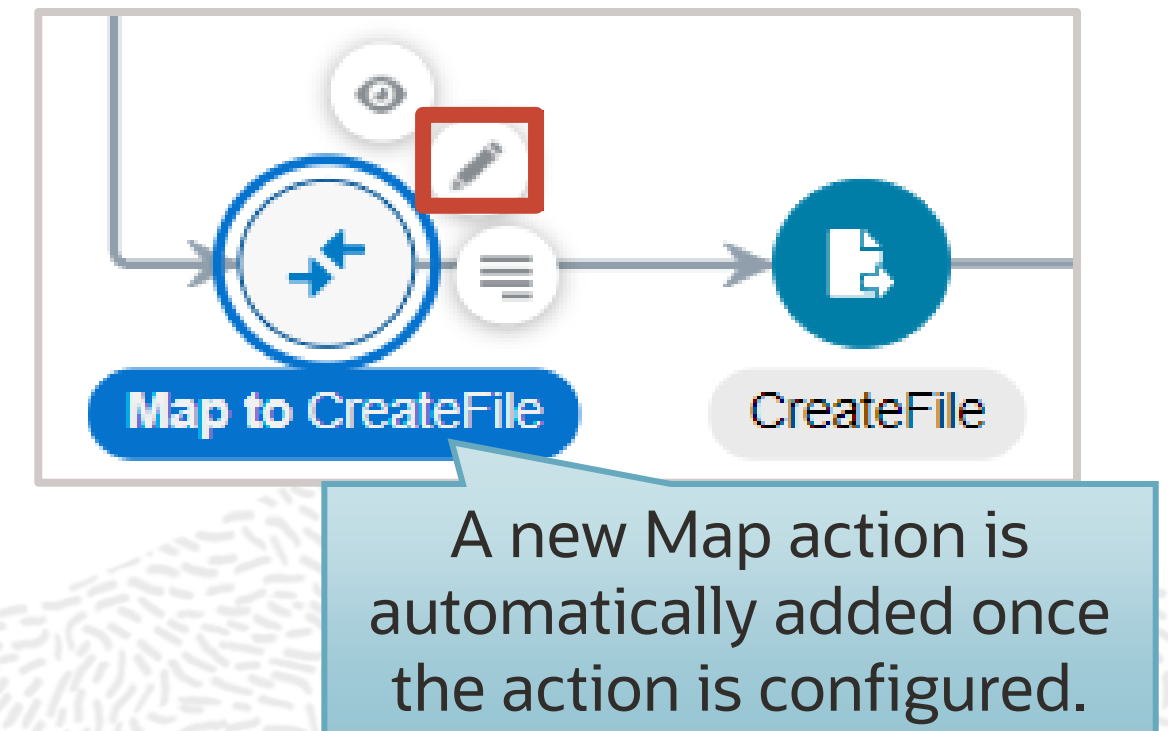
* Choose Stage File Operation: Write File

* Specify the File Name: `concat($ICSFile/nsmpr2:ICSFile/nsmpr2:Properties/nsmpr2:filename, ".csv")`

* Specify the Output Directory: `/output_directory`

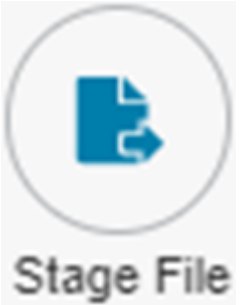
Append to Existing File: ☐

Use the Expression Builder to define the file name.



Edit the **Target** in the mapper to define the contents of the file to be created or appended.

File Schema Options



For **Read Entire File**, **Read File in Segments**, and **Write File** operations, you are required to specify a schema to define the structure of the file:

- Select how you will describe the file structure on the ***Format Definition*** page.
 - To be exposed within the integration data mappers as an XSD structure

A screenshot of a configuration window titled "Configure the Stage File Action with Schema Options" with the subtitle "Define the schema options for Stage File Operation". The window has a sidebar on the left with a navigation menu containing: "Basic Info" (with a green checkmark), "Configure Operation" (with a green checkmark), "Schema Options" (highlighted in blue), "Format Definition", and "Summary". The main area of the window contains two questions. The first question is "Do you want to specify the structure for the contents of the file?" with radio buttons for "Yes" (selected) and "No". The second question is "Which one of the following choices would be used to describe the structure of the file contents?" followed by a dropdown menu. The dropdown menu is open, showing the following options: "Sample delimited document (e.g. CSV)" (selected and highlighted in blue), "XML schema (XSD) document", "Sample XML document (Single or No NameSpace)", and "Sample JSON document".

Configure the Format Definition (Review)



On the **Format Definition** page:

- If providing a sample CSV file, a new schema is created based on the file structure
- If providing a sample JSON or XML file, select the schema element to be used
- If providing an XSD file, select the schema element:
 - It can be used for processing JSON or XML files.
 - If a native schema (.nxsd) is provided, it can be used for processing CSV or any other native file formats (such as fixed length, complex types, etc.)

✓ Basic Info

✓ Configure Operation

✓ Schema Options

Format Definition

Summary

Create a New Schema from a CSV file

Select a New Delimited Data File

Choose File

 No file chosen

Selected File Name newEmployeeFile.csv

* Enter the Record Name

For example, Address

* Enter the Recordset Name

For example, AddressBook

Select the Field Delimiter

Comma (,) ▾

Character Set

UTF8 ▾

Optionally Enclosed By

" ▾

Terminated By

\$(eol) ▾

☒ Detach

☐ Use First Row as Column Headers

☐ Mark All As Optional

C1	C2	C3	C4	C5
<div>String ▾</div>	<div>String ▾</div>	<div>String ▾</div>	<div>String ▾</div>	<div>String ▾</div>
<div>Mandatory ▾</div>	<div>Mandatory ▾</div>	<div>Mandatory ▾</div>	<div>Mandatory ▾</div>	<div>Mandatory ▾</div>
Salutation	FirstName	LastName	CorrespondenceL...	PersonNu

Provide a sample JSON document from the file system

Select a New File

Choose File

 No file chosen

Selected File Name ExternalCountryResponse.json

Select the Schema Element

request-wrapper ▾

Provide a sample XML document from the file system.

XML document should contain no namespace or only single namespace.

? Select a New File

Choose File

 No file chosen

Selected File Name persons.xml

Select the Schema Element

allpersons ▾

Select an existing XML schema or schema archive from the file system.

Schema archive can have a single top level schema with nested imports and includes containing absolute or relative paths.

Select a New File

Choose File

 No file chosen

Selected File Name eIWO_target.nxsd

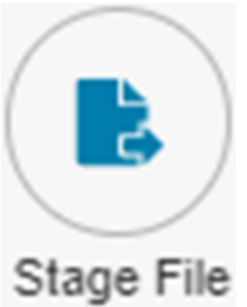
Select the Schema Element

ElectronicIncomeWithholdingOrders ▾

For more information about native file formats, refer to the lesson titled “File Handling Concepts and Options” in this course.



Stage File: List Files



Return the list of files that are located in a specified VFS directory

?

* Choose Stage File Operation

List Files

* Specify the Directory to List Files from

\$DownloadFile/nsmpr5:DownloadFileToICSResponse/nsmpr1:DownloadResponse/nsmpr1:ICSFiles/nsmpr2:ICSFile/nsmpr2:Properties/nsmpr2:directory

✎

Use the Expression Builder to define the file name.

?

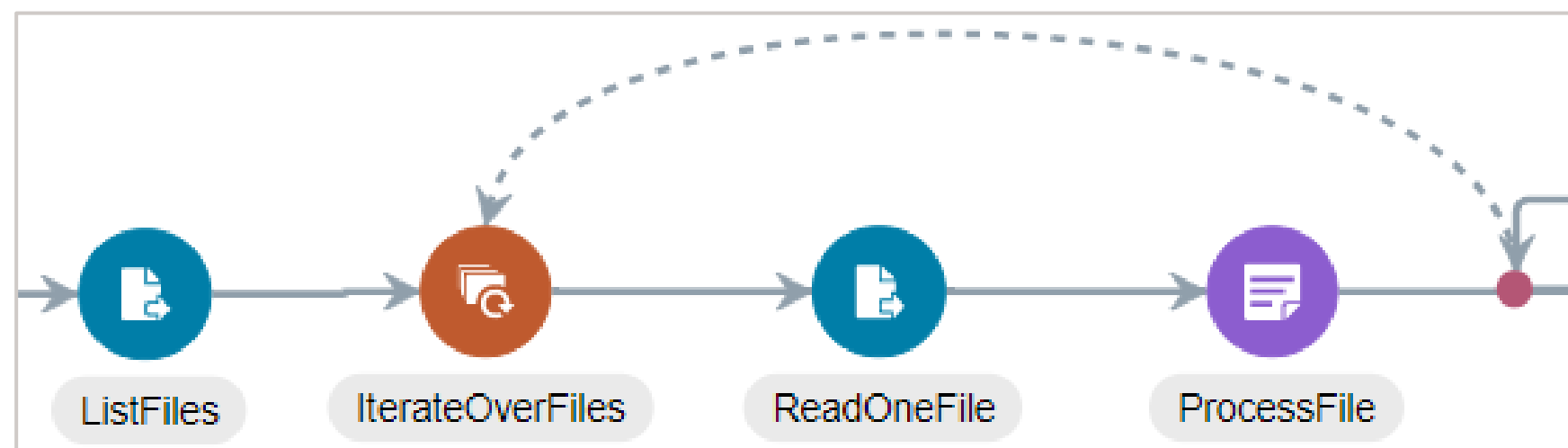
* Specify the File Pattern to use

**

List Files Recursively

☐

Check to list subdirectory files recursively.



Adds a Source element that can be referenced in the next action.

Source

Find...

▶ <> \$ICSFile

▲ <> \$ListFiles

▲ <> *ListFilesResponse

▲ <> *ListFilesResponse

▲ <> *FileList

▲ <> ICSFile

<> *FileReference

▲ <> *Properties

<> *filetype

<> *directory

<> *filename

<> *lastModifiedTime

<> *creationTime

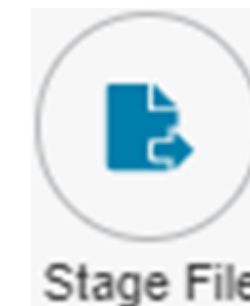
<> *size

<> *checksum

For example:
Add a for-each loop to subsequently process each file in the list.



Stage File: Unzip File / Zip Files



Unzip File: Extract an archived file to a specified VFS directory.

Zip Files: Archive all files within the specified VFS directory to another VFS directory.

?

* Choose Stage File Operation

Unzip File

Configure File Reference

No

Yes

* Specify the Zip File Name

\$DownloadFile/nsmpr5:DownloadFileToICSResponse/nsmpr1:DownloadResponse/nsmpr1:ICSFiles/nsmpr2:ICSFile/nsmpr2:Properties/nsmpr2:filename

* Specify the Zip File Directory

\$DownloadFile/nsmpr5:DownloadFileToICSResponse/nsmpr1:DownloadResponse/nsmpr1:ICSFiles/nsmpr2:ICSFile/nsmpr2:Properties/nsmpr2:directory

* Specify the Directory to Unzip

"/extracted_files"

Use the Expression Builder to specify the filename & directory...

unzipFile

?

* Choose Stage File Operation

Unzip File

Configure File Reference

No

Yes

* Specify the File Reference

\$DownloadFile/nsmpr5:DownloadFileToICSResponse/nsmpr1:DownloadResponse/nsmpr1:ICSFiles/nsmpr2:ICSFile/nsmpr2:FileReference

* Specify the Directory to Unzip

"/extracted_files"

...or the file reference.

Specify a destination VFS directory

* Choose Stage File Operation

Zip Files

* Specify the File Name

concat("data_updated",".zip")

* Specify the Directory to zip

"/processed"

* Specify the Output Directory

"/"

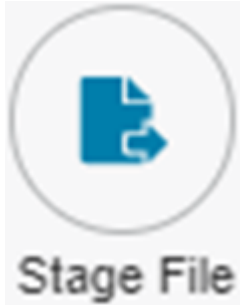
Specify a name for the new archive file.

Location of staged files to be archived

zipFile



Stage File: Encrypt File / Decrypt File



Encrypt File: Select the appropriate uploaded PGP public key to create an encrypted file from a specified file reference.

Decrypt File: Select the appropriate uploaded PGP private key to create a decrypted file from a specified file reference.

Certificates

Upload

6 Certificates

Installed By : User

Name	Type	Category	Status
demo2_private <div>EXPIRES IN 79 YEARS</div>	PGP	Private	Configured
demo2_public <div>EXPIRES IN 79 YEARS</div>	PGP	Public	Configured
demo1_private <div>EXPIRES IN 79 YEARS</div>	PGP	Private	Configured
demo1_public <div>EXPIRES IN 79 YEARS</div>	PGP	Public	Configured

Choose Stage File Operation

Encrypt File

Specify the File Reference

Click on Expression Builder to choose file reference

Specify the File Name

Click on Expression Builder to choose file

Specify the Output Directory

Click on Expression Builder to choose directory

Select PGP Key to encrypt file

Select PGP Key to encrypt

demo1_public

demo2_public

Choose Stage File Operation

Decrypt File

Specify the File Reference

Click on Expression Builder to choose file reference

Specify the File Name

Click on Expression Builder to choose file

Specify the Output Directory

Click on Expression Builder to choose directory

Select PGP Key to decrypt file

Select PGP Key to decrypt

demo2_private

demo1_private



Agenda

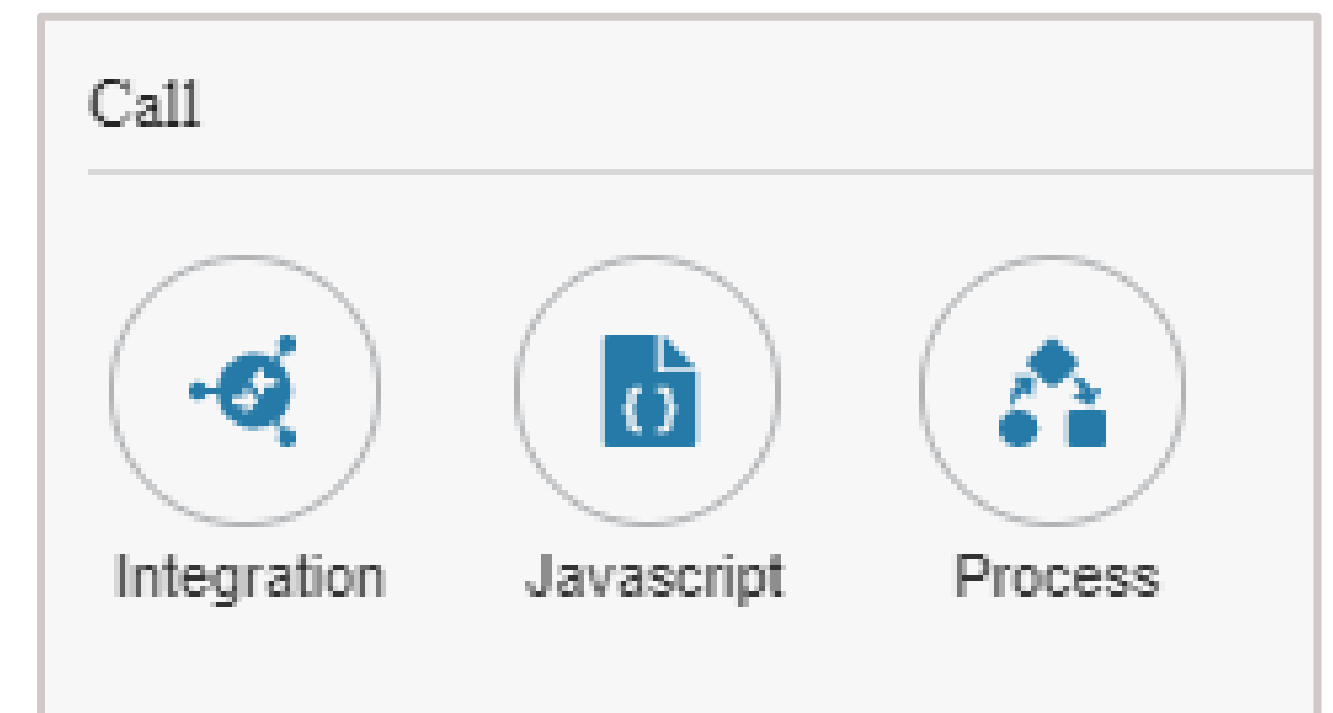
- Orchestration Design Canvas Editing Options
- Orchestration Menu Actions
 - Data Section
 - Call Section
 - Collection Section
 - General Section
 - End Section



Call Section

Selecting the **Actions** menu provides the following functionality in the **Call** section:

- **Integration:** Provides a wizard for configuring a call to an invokable integration activated within the same OIC instance
- **Javascript:** Provides a wizard for configuring a call to a registered library function
- **Process:** Provides a wizard for configuring a call to an invokable process within a Process Application deployed to the same OIC instance

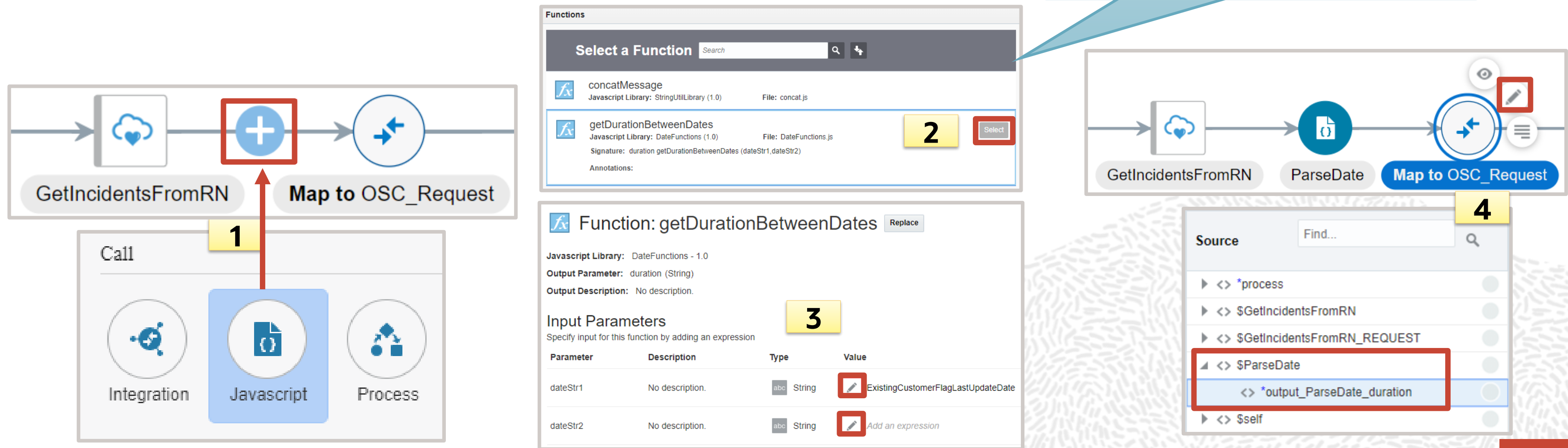


Invoking JavaScript Functions (Review)



- 1. Drag the **Javascript** call action to the integration flow.
- 2. Select the custom function.
- 3. Use the Expression Builder to define parameter values.
- 4. The return value is now available as a Source data element.

All Library functions that have been classified as **Orchestration** will appear in the list of callable functions.



Invoking Local Integrations



Using the **Integration** call action:

- Eliminates the need to create any explicit connection for that endpoint
- Optimizes the call in-memory without traversing the network stack
- Can invoke a scheduled orchestration from another integration
- Supports the best practice pattern of breaking a single large integration into multiple smaller integrations
 - Promotes better design and provides modular functionality for easier maintainability

Local Integration Selection Wizard

1. Add the **Integration** action to the integration to launch the wizard.
2. Provide a name for the call on the **Basic Info** page.
3. Select an active integration.
4. Select the operation.
5. Review and click **Done**.
6. Edit the mapper to configure **Target** data.

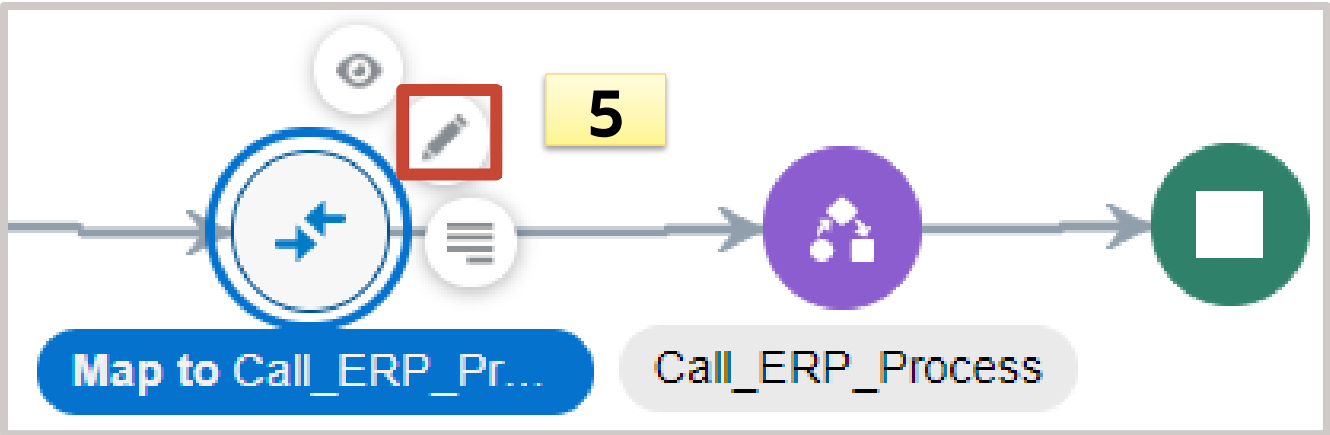
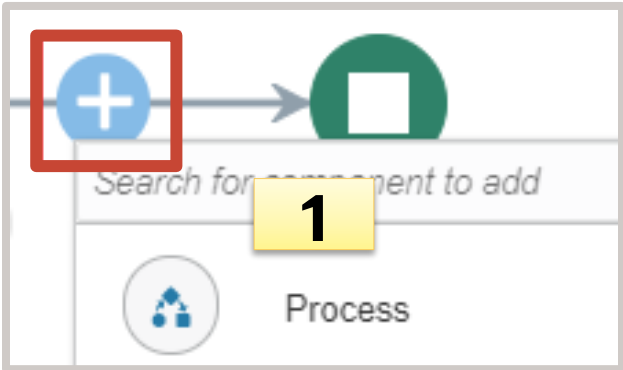
The collage illustrates the steps of the Local Integration Selection Wizard:

- Step 1:** A screenshot of a flow diagram where a blue square icon with a plus sign is added to a flow. A search bar below it shows 'Integration' selected, with 'Javascript' as an alternative.
- Step 2:** The 'Welcome to the Local Integration Selection Wizard' dialog box. It shows the 'Basic Info' tab with the operation name 'CallHelloWorld' and a description 'What does this local integration invocation do? Call Hello World'.
- Step 3:** The 'Select a Local Integration' dialog box. It shows the 'Select Integration' tab with a list of integrations. 'Hello World (1.2.0)' is selected, highlighted with a red box.
- Step 4:** The 'Select the required Operation' dialog box. It shows the 'Select Operation' tab with a list of operations. 'GET' is selected, highlighted with a yellow box.
- Step 5:** The 'Local Integration Selection Summary' dialog box. It shows the 'Summary' tab with details for the 'CallHelloWorld' integration, including the integration name, endpoint name, resource name, and operation.
- Step 6:** A screenshot of the flow diagram showing the 'Map to CallHelloWorld' action and the 'CallHelloWorld' action.



Invoking a Local OIC Process

- 1. Add the **Process** action to the integration to launch the wizard.
- 2. Provide a name for the call on the **Basic Info** page.
- 3. Select the Process Application, and then the message-based process.
- 4. Review and click **Done**.
- 5. Edit the mapper to configure **Target** data.



Welcome to the Process Selection Wizard
This wizard guides you through the selection of a process from Integration Cloud.

Basic Info

Select Process

Summary

* What do you want to call your process invocation?
Call_ERP_Process

What does this process invocation do?
Describe the process's purpose and detail

Select a Message Based Process
Select a message based process from an application workspace, and choose an operation(if more than one available).

Basic Info

Select Process

Summary

* Application
ERP Example

* Process (Message Based Pattern)
ERP Example
Enter Process Name
Process | Revision 1.0 (Default)

Select Process

Process Selection Summary
Process selected successfully.

Basic Info

Select Process

Summary

Application Name ERP Example

Process Name Process

Operation Name start

Done



Agenda

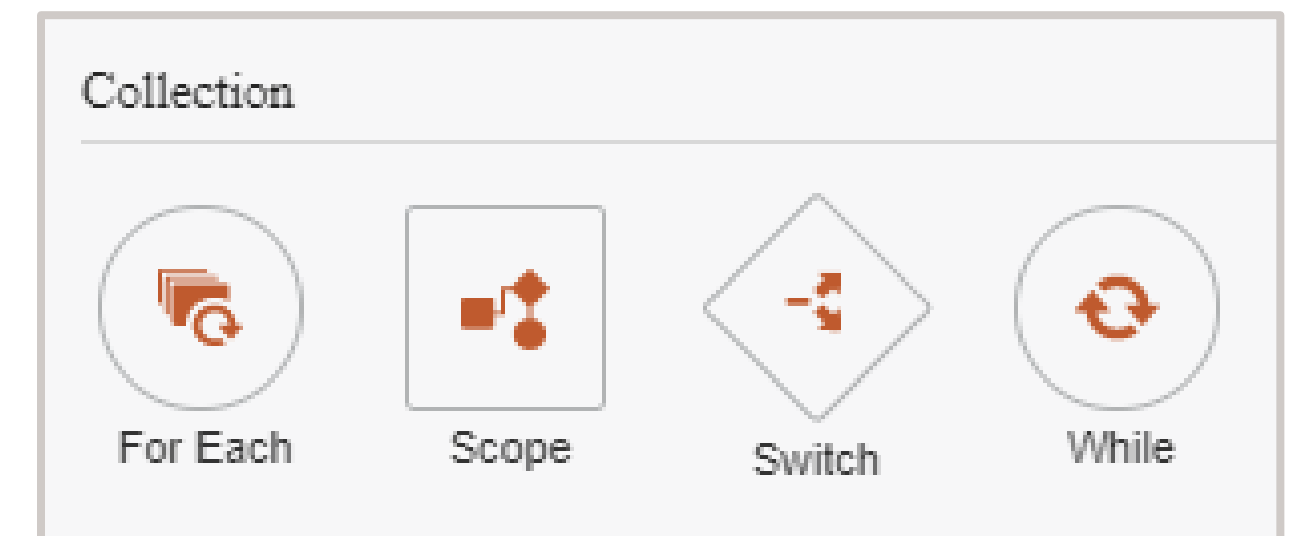
- Orchestration Design Canvas Editing Options
- Orchestration Menu Actions
 - Data Section
 - Call Section
 - Collection Section
 - General Section
 - End Section



Collection Section

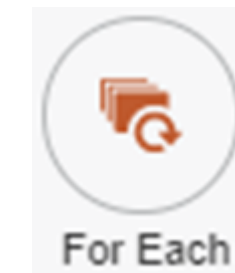
Selecting the **Actions** menu provides the following functionality in the **Collection** section:

- **For Each:** Loop over a repeating element and execute one or more Actions or Invokes.
- **Scope:** It provides a context for child elements.
 - Most often used for defining fault handlers
- **Switch:** Add one or more routing expressions to define switch branches within the Integration flow.
- **While:** Loop over and execute one or more Actions or Invokes until condition is met.



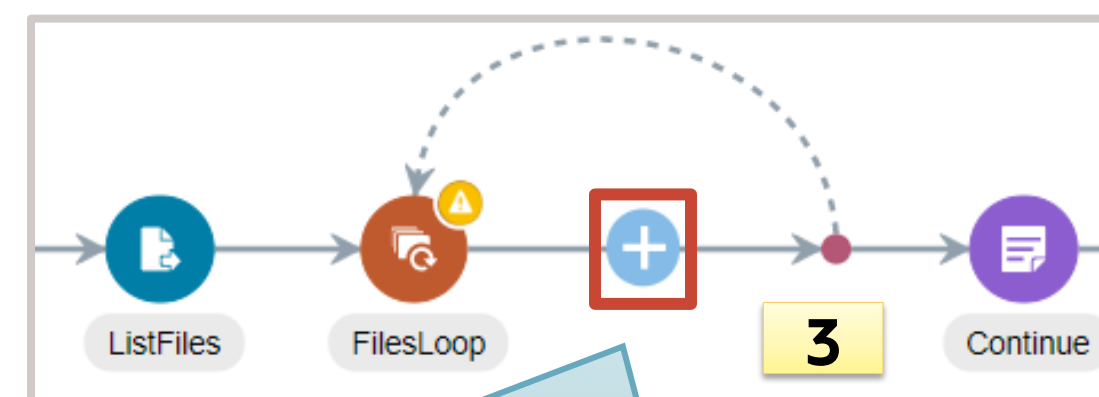
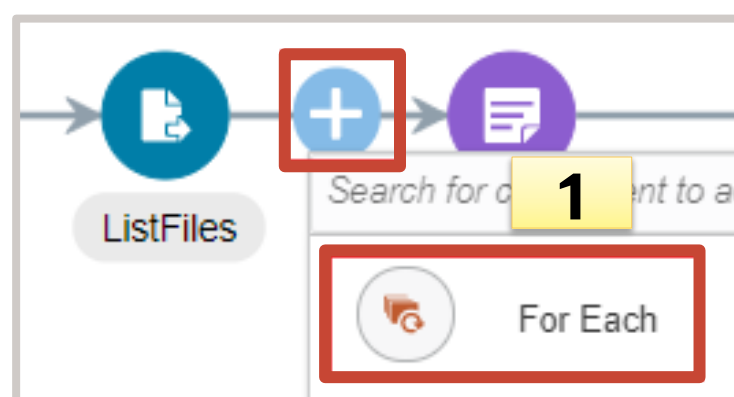
We'll defer coverage of **Scope** to Lesson 12: *Orchestration Scopes & Fault Handling*.

For-Each Loop



The **For-Each** action uses a repeating element to indicate the number of iterations through the loop.

- The loop provides a scope for creating activities and variables.
- Nested repeating elements can be handled with nested For-Each actions as needed.



Create Action

For Each

View Filter Detach

Source Find...

For Each action enables iterating over a repeating element and executing one or more activities within its contained scope. Specify the repeating element by selecting from the source tree on the left and provide a name for the current element. The current element can be used for activities within the For Each scope. Select Process items in parallel if elements required to be processed concurrently.

* Name FilesLoop

Description Enter a brief description...

* Repeating Element \$ListFiles/nsmpr6:ListFilesResponse/nsmpr8:ListFilesResponse/nsmpr8:st/nsmpr2:ICSFile

* Current Element Name NextFile

Process items in parallel ☐

2

Add one or more actions within the loop.

Current Element Name will be visible as a **Source** for subsequent mappings.

Sequential or Parallel

Source Find...

*schedule

\$DownloadFile

\$DownloadFile_REQUEST

SFTPDownloadDir

\$FileToDownload

\$ListFiles

\$NextFile

*ICSFile

*FileReference

*Properties

*filetype

*directory

*filename

*lastModifiedTime

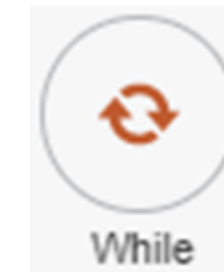
*creationTime

*size

*checksum

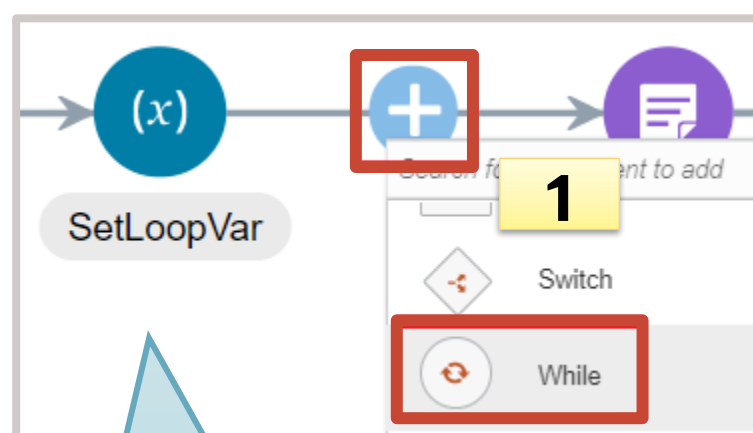


While Loop

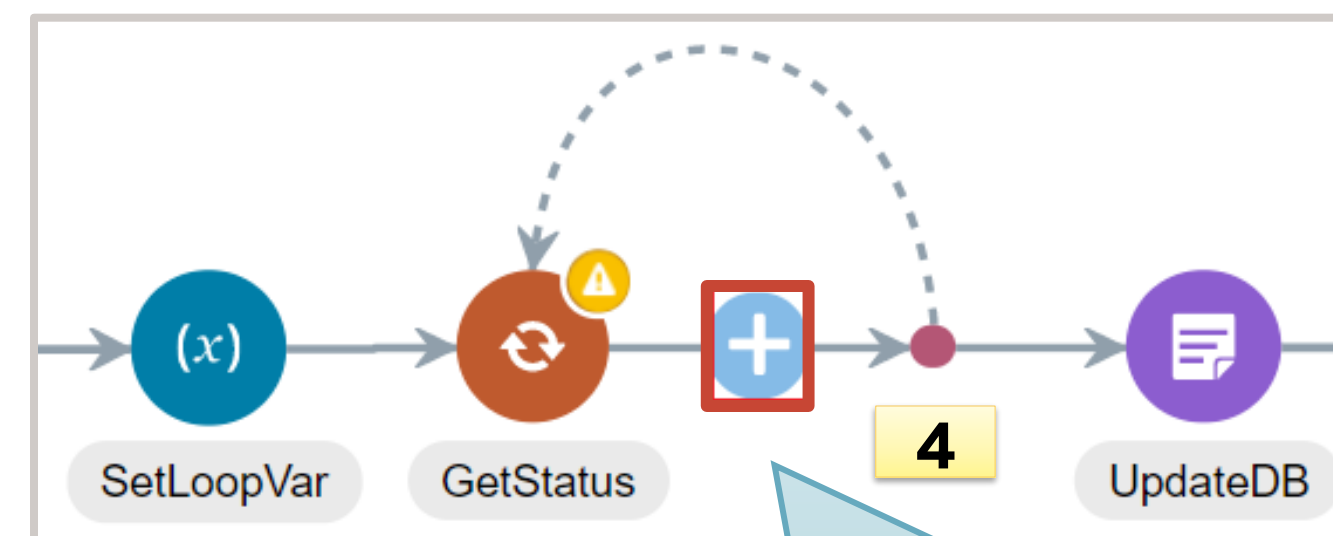


The **While** action iterates the loop based on the value of an execution condition.

- The loop provides a scope for creating activities and variables.
- Logic inside the loop must provide an update to the loop condition in order to exit the loop.



Normally, you initialize the loop variable in advance.



Add one or more actions within the loop.

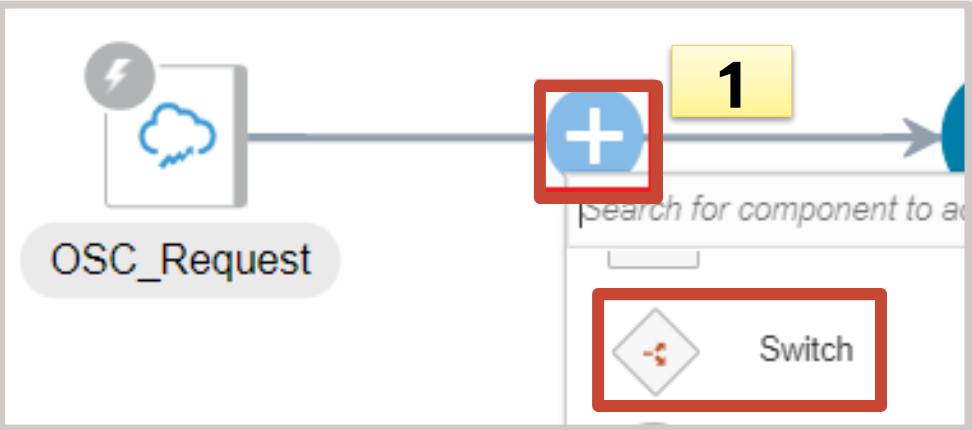
Define the condition to evaluate at the start of each loop.



Switch Branches



Use the **Switch** action to add conditional routing expressions within the integration or to add conditional logic such as *If* and *If-Else*.



Click to define the condition.

undefined

Expression in "WhileLoop (1.0)"

Inputs

View Filter Detach

Source Find...

AddressLine4

Building

City

Country

3

Expression Name

Provide a name or short description for the expression.

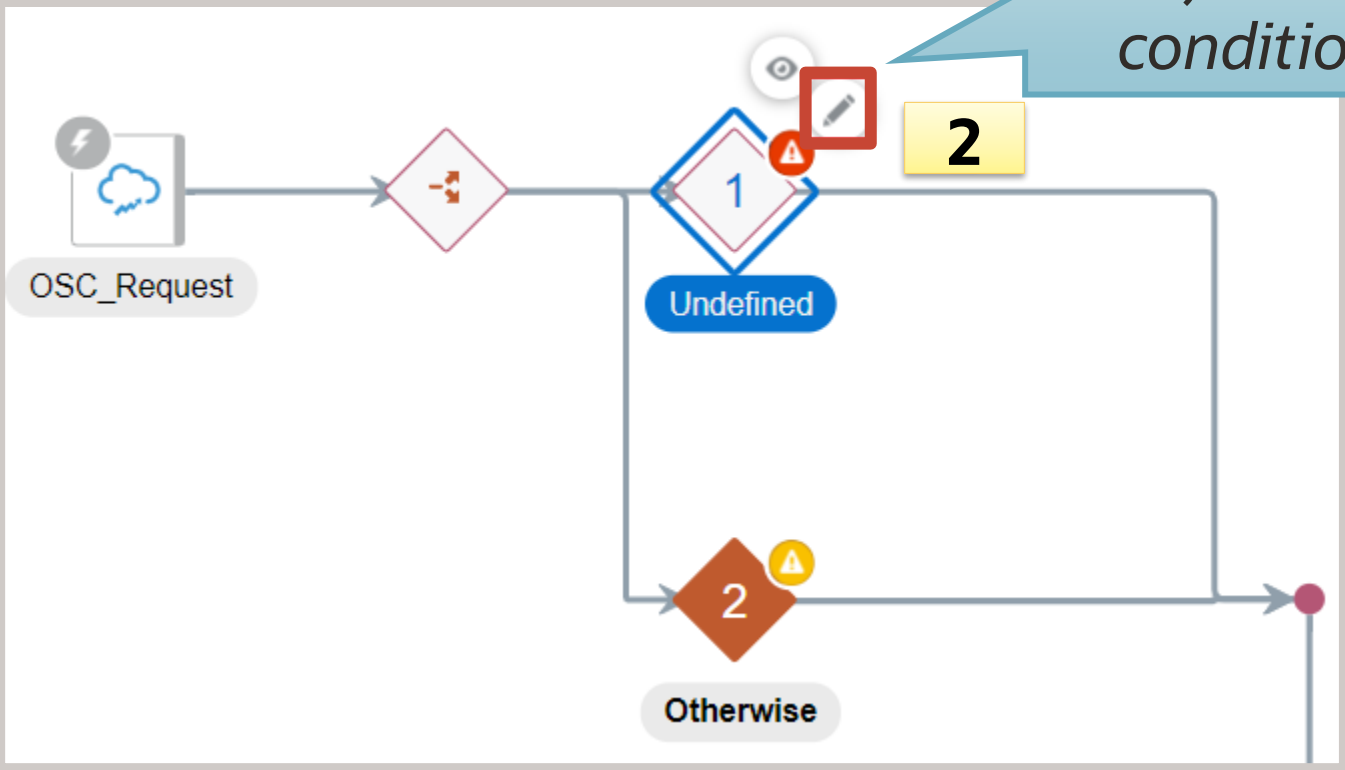
Expression

/nssrcmpr:process/nssrcmpr:Account/nsmpr9:Country = "USA"

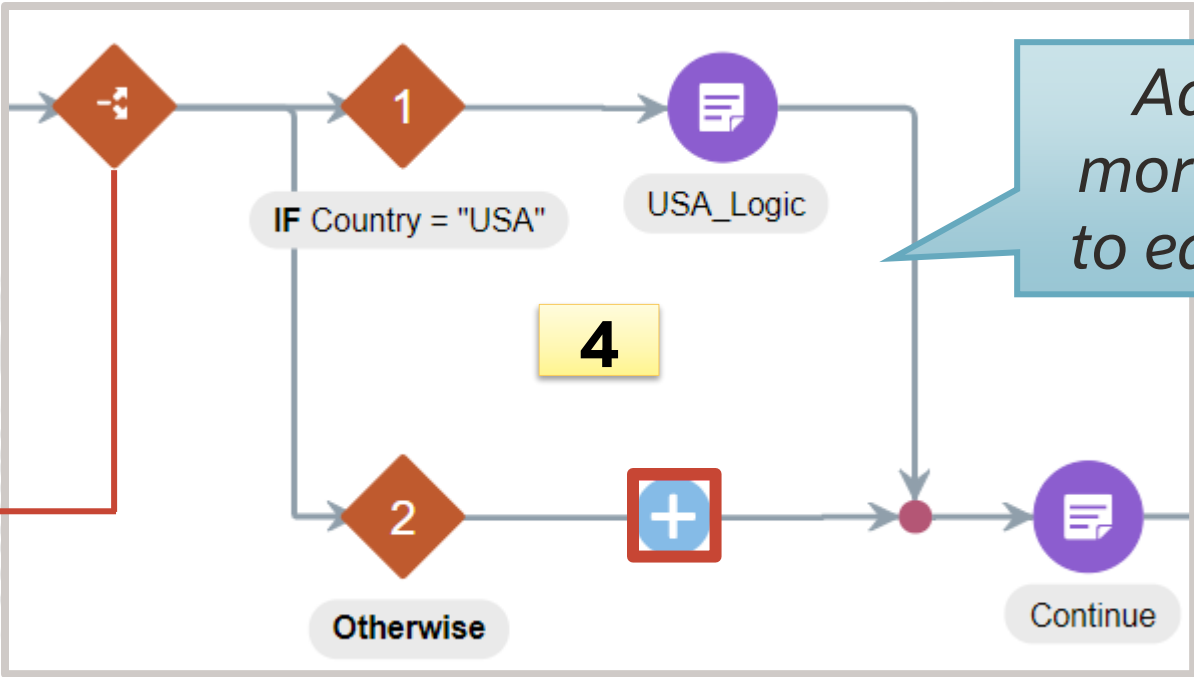
Expression Summary

Country = "USA"

Use the Expression Builder for each undefined branch condition.



Optionally, add another branch.



Add one or more activities to each branch.



Agenda

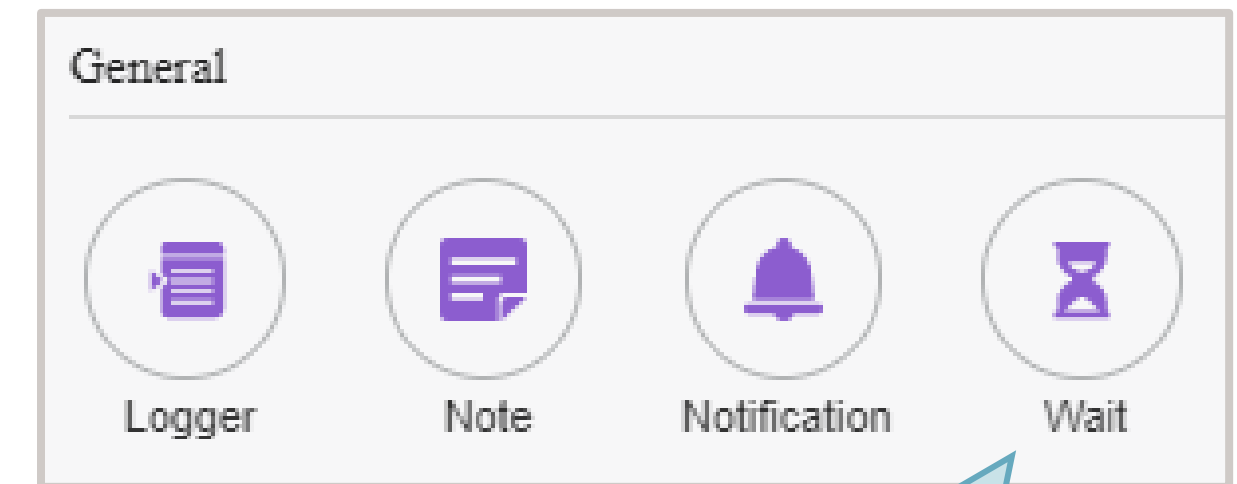
- Orchestration Design Canvas Editing Options
- Orchestration Menu Actions
 - Data Section
 - Call Section
 - Collection Section
 - General Section
 - End Section



General Section

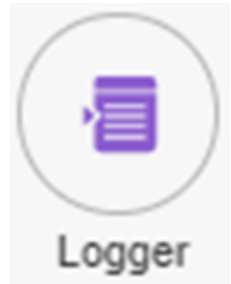
Selecting the **Actions** menu provides the following functionality in the **General** section:

- **Logger:** Logs messages to the activity stream and diagnostic log
- **Note:** Is a design-time placeholder for developer notes
- **Notification:** Sends an email message to one or more recipients
- **Wait:** Provides a delay to the execution of an Integration flow for a specified period of time



The Wait action is not available in synchronous integration flows.

Logger Action



You can log messages to the Activity Stream at any point in the integration.

- Select to log *Always* or *When trace is enabled*.
- Create a static message or a dynamic message based on **Source** data or variables by using the Expression Builder.

1 Click the '+' button in the component palette to add a new action.

2 In the 'Create Action' dialog, enter a unique name (e.g., 'LogData') and a description.

3 In the 'Log' section, select the logging mode: ☐ Always or ☒ When trace is enabled.

4 Click the edit icon (pencil) next to the 'Logger Message' field to open the Expression Builder.

5 In the 'Log Message' configuration window, define the expression to log. The expression is built from the 'Source' data (e.g., 'Greeting') using functions like 'concat'.

Log Message Configuration:

- Inputs: View, Filter, Detach
- Source: *schedule, startTime, SCall, *GreetingResponse, Greeting
- Expression: `concat("Processing greeting: ", $Call/nsmpr0:GreetingResponse/nsmpr0:Greeting)`
- Expression Summary: `concat("Processing greeting: ", Greeting)`

LogData Test2 (1.0)

Configure Logger

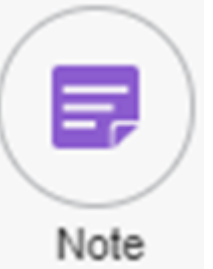
A Logger action enables you to log messages and variables to the Activity Stream at any point in the integration flow.

Log: ☐ Always, ☒ When trace is enabled

* Logger Message: Specify what message should display in the Activity Stream. You specify the message by adding an expression. You can define a static message or include variables to be logged with the message.

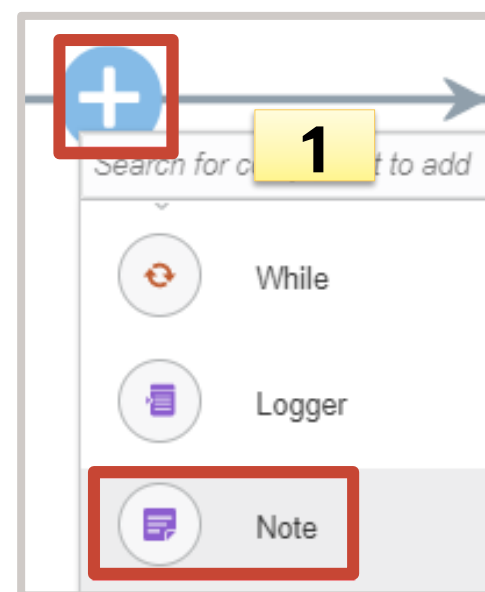
LogData

Note Action



The **Note** action is a design-time placeholder used as a reminder for later development activity or used for collaboration comments.

- If left within the integration flow at activation, there is no impact because these actions are skipped at run time.



Create Action

Note

Please give a unique name and description to this action. Please give a name, description.

* Name

UpdateDB

* Description

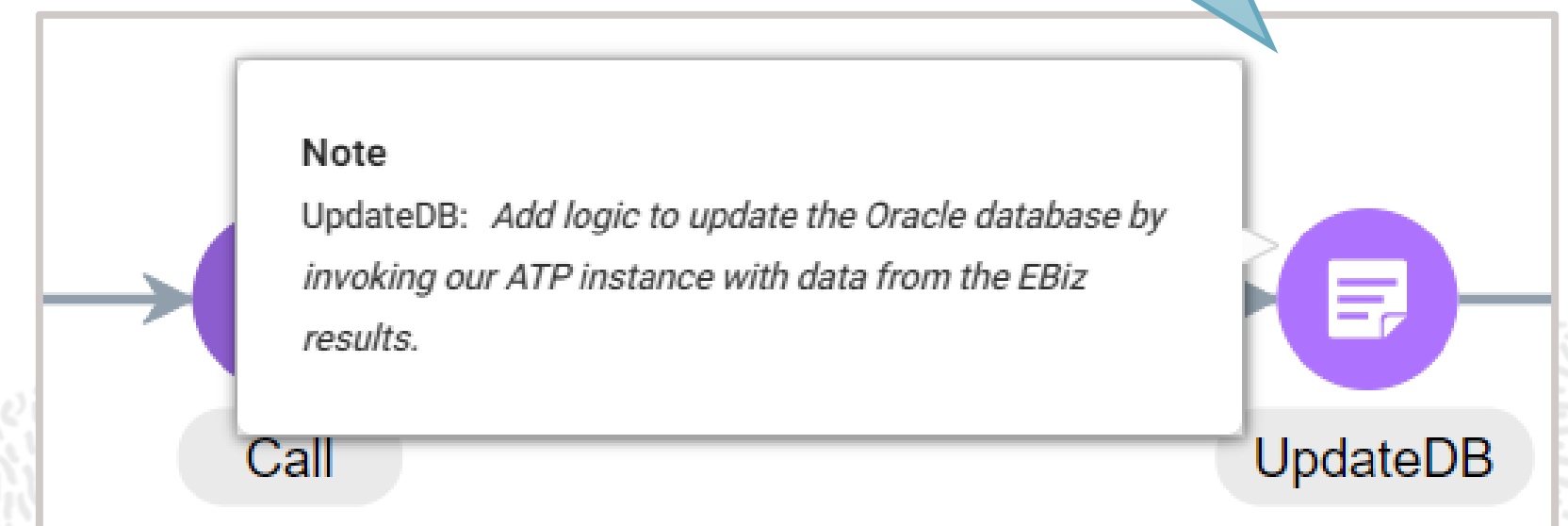
Add logic to update the Oracle database by invoking our ATP instance with data from the EBiz results.

2

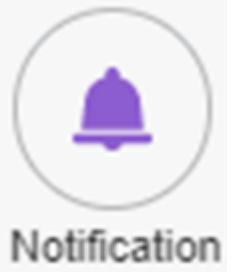
OK

Cancel

Hover the mouse over the Note icon to view the text.

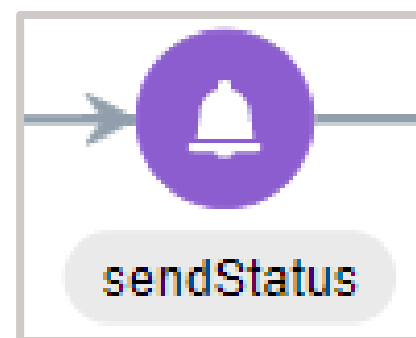
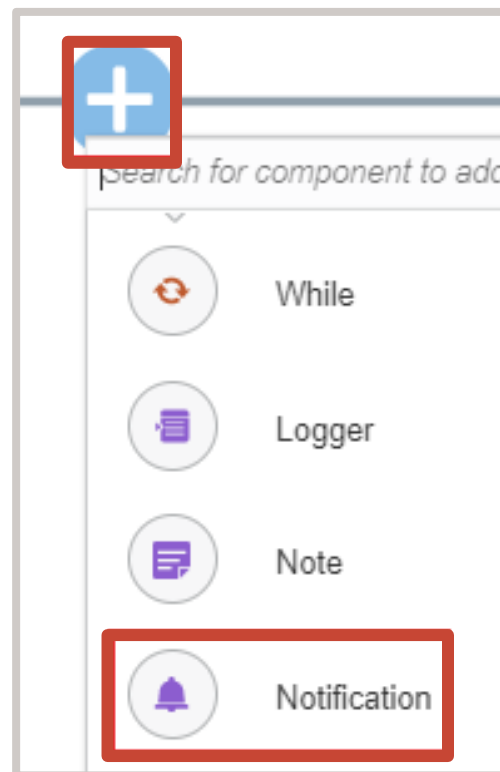


Notification Action




Use the **Notification** action to send messages to one or more email addresses.

- Send HTML or plain text in the Body. Add file attachments up to 2 GB.



sendStatus ForEachExample (1.0) Close Validate


Notification

* From: "oic_admin@example.com" 


* To: \$emailTo


* Subject: concat("ERP Import Status - ", startTime)

* Body: The status of the ERP import is {import_status}.

Attachments 

FileReference

Parameter Name: import_status Description: Status returned from ERP import Parameter Value: result 

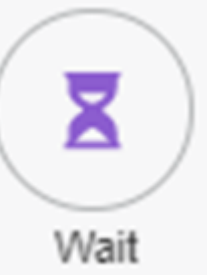


Use the Expression Builder for defining values.

Add 1 or more file attachments

Define local parameters to provide dynamic content in the Body.

Wait Action



Use a **Wait** action to define a static execution delay.

- This action is not available in App Driven Orchestration style integrations that are configured with a synchronous trigger (*request-response message exchange pattern*).
- Maximum: 6 hours

+

Search for component to add

Logger

Note

Notification

Wait

Create Action

Wait

Please give a name, description and set time for this wait action. Maximum wait time should be 6 hrs.

* Name

Wait_5_Mins

Hours

0

Minutes

5

Seconds

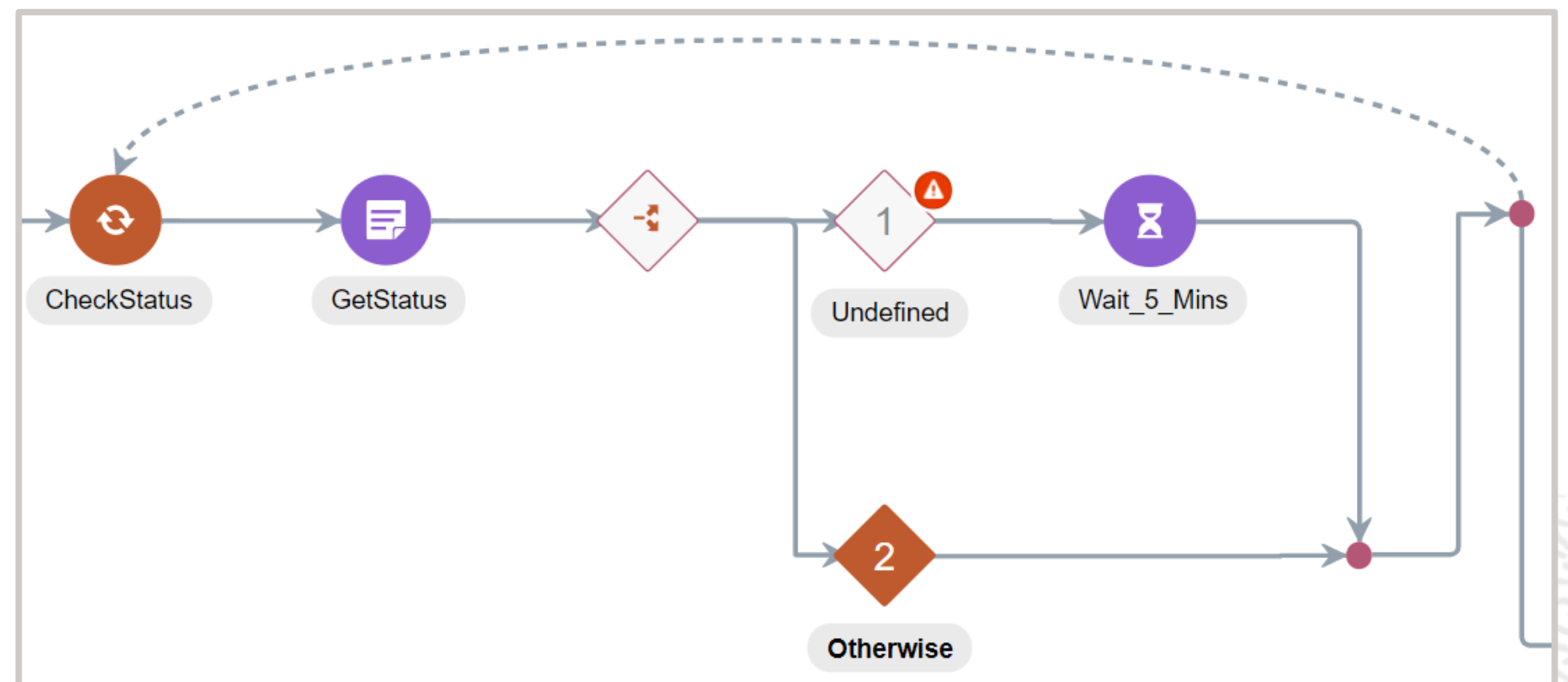
0

Description

Wait for 5 minutes prior to the next iteration.

Create

Cancel



Agenda

- Orchestration Design Canvas Editing Options
- Orchestration Menu Actions
 - Data Section
 - Call Section
 - Collection Section
 - General Section
 - End Section



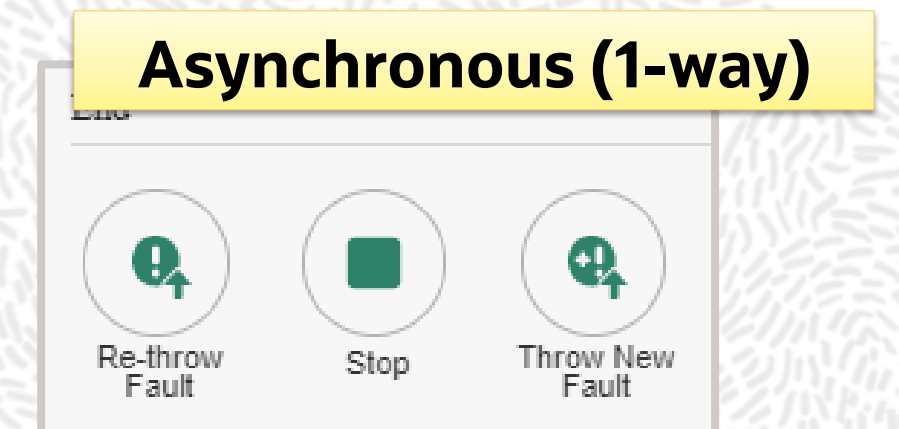
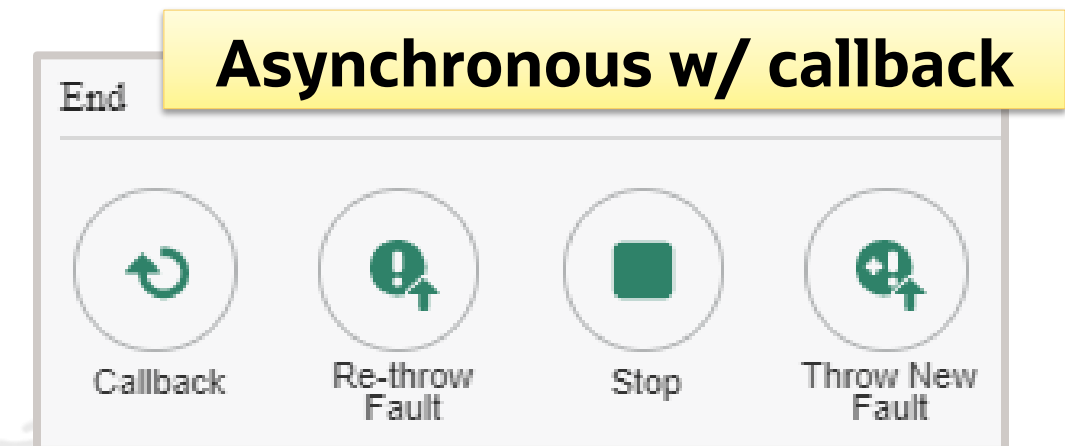
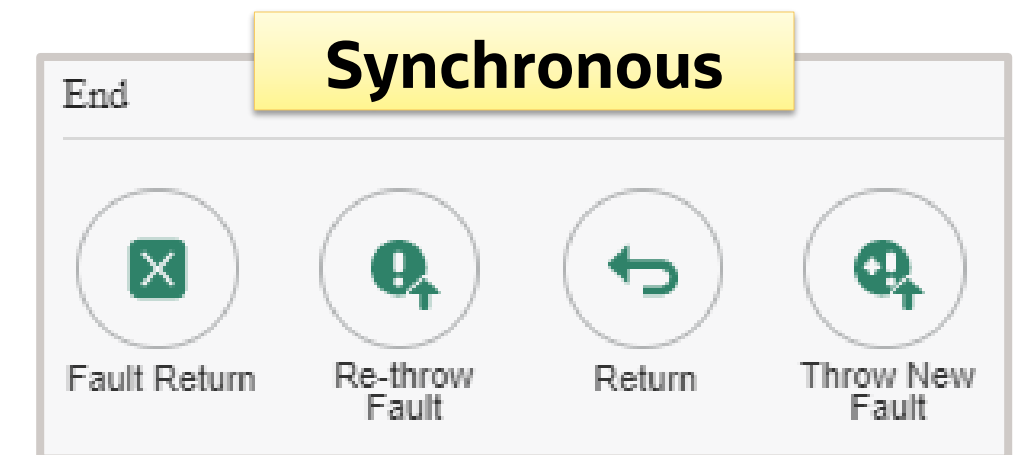
End Section

Selecting the **Actions** menu provides the following functionality in the **End** section:

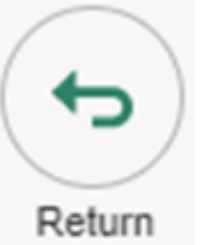
- **Re-throw Fault:** Used only within a fault handler
- **Throw New Fault:** Creates and throws a new fault to the next fault handler
- **Fault Return:** Executes an immediate fault response to the client application
- **Return:** Executes an immediate response to the client application
- **Callback:** Executes a callback invocation to the client application
- **Stop:** Ends integration flow instance processing

We'll defer coverage of Fault actions to Lesson 12: *Orchestration Scopes & Fault Handling*.

*The available **End** actions will depend on the integration's message exchange pattern.*

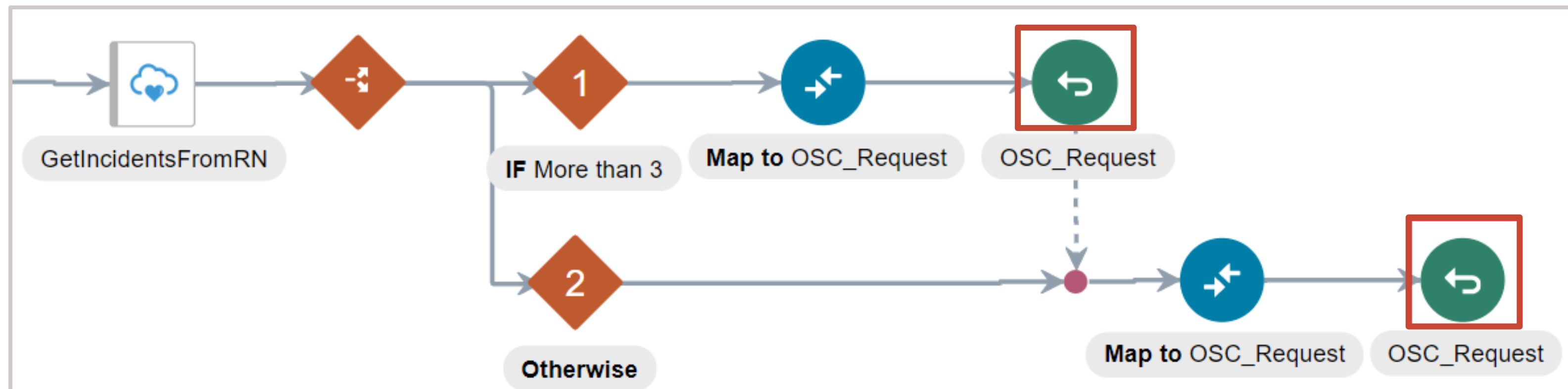


Return Action

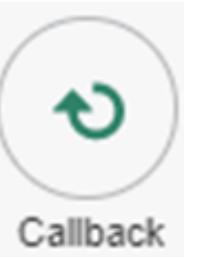


An additional **Return** action could be added as dictated by the dynamic logic of the business use case.

- This action is available only in synchronous integrations.

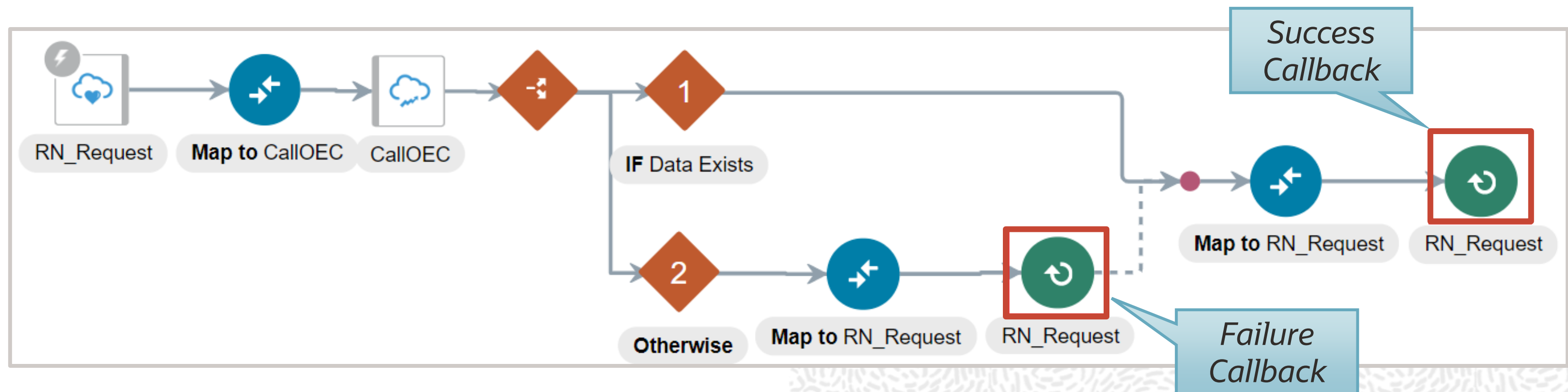


Callback Action

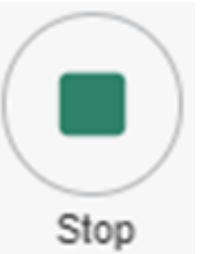


A **Callback** action can be added to provide one or more additional callbacks as dictated by the business use case and specified interface.

- This action is available only in App Driven Orchestration style integrations that define an asynchronous w/callback trigger interface.

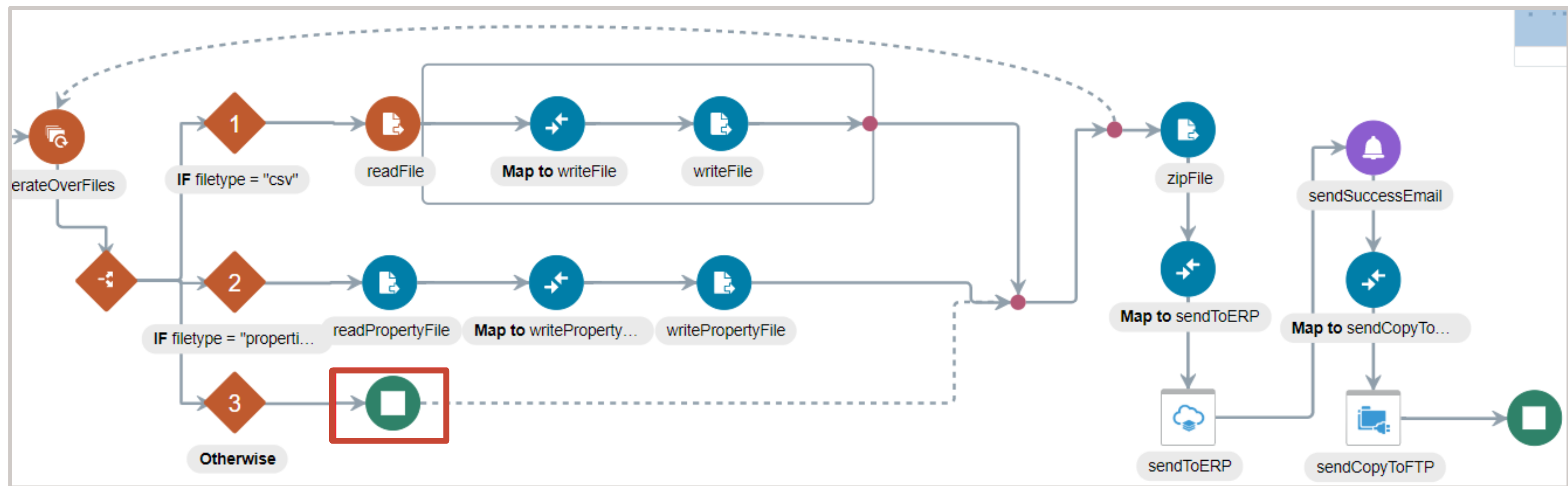


Stop Action



A **Stop** action could be added to a conditional branch to end the integration instance earlier as dictated by the runtime logic of the use case.

- This action is not available in synchronous integrations.

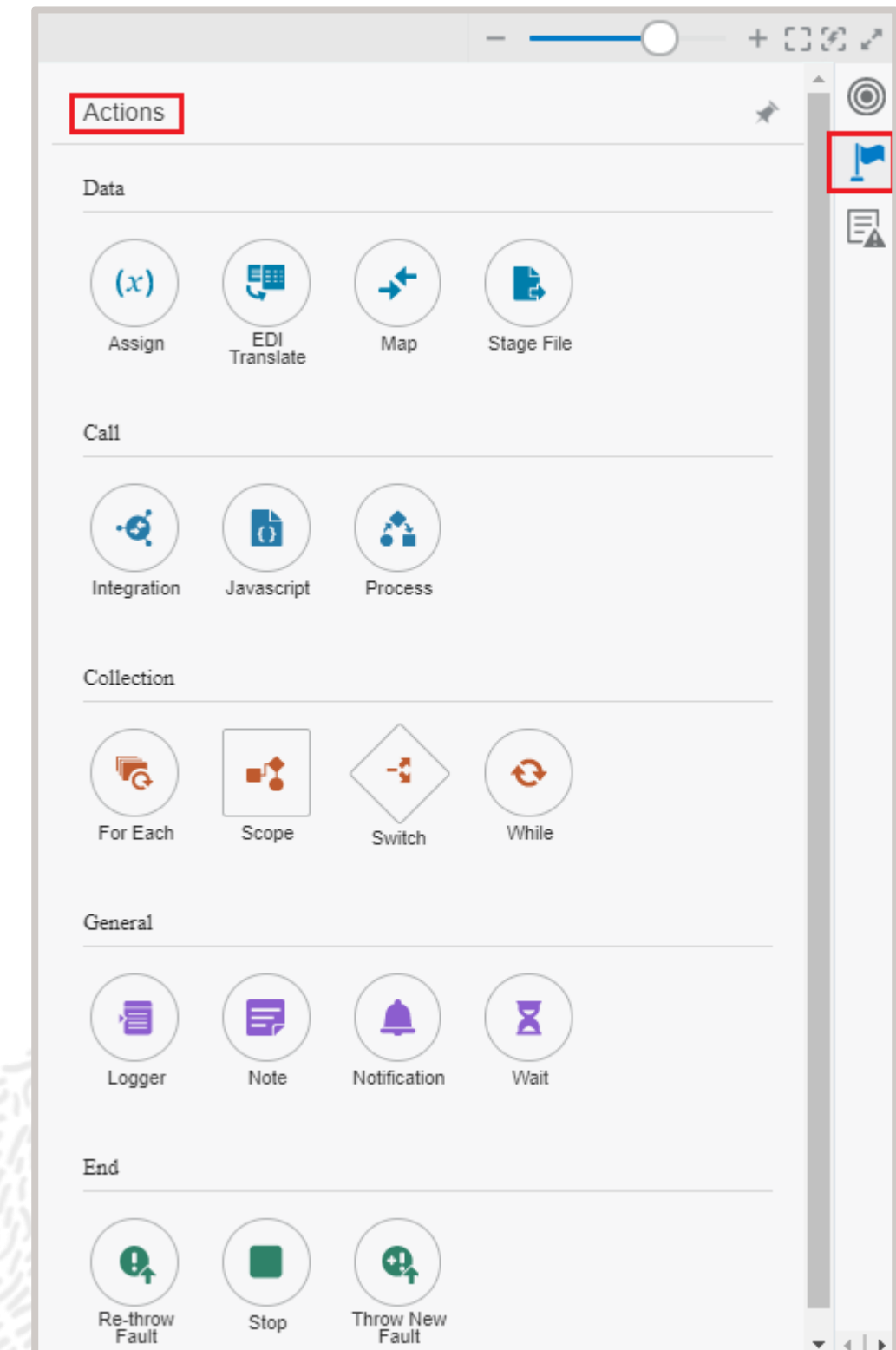


Actions Menu (Review)

Most actions can be placed anywhere within the integration flow at any of the following locations:

- The main flow path
- The Global fault handler
- For-Each loops
- While loops
- Branches of a Switch action
- Scopes
- Scope fault handlers

More coverage will be provided in
“Orchestration Scopes and Fault Handling”
 later in this course.



Summary

In this lesson, you should have learned how to:

- Leverage the design canvas view options
- Use actions in orchestration style integrations
- Be familiar with Stage File action operations
- Invoke OIC resources including local integrations and processes
- Design with looping constructs and conditional branches



Practice 9-1: FTP, Data Mapping, and HCM Cloud

This practice includes:

- PART 1 – Clone and Edit a New Integration Flow
- PART 2 – Implement Additional Custom Data Mapping
- PART 3 – Activate and Run the Integration
- PART 4 – Exploring the Integration Flow Implementation Details

